MODIFICATION OF THE TASDA COMPUTER PROGRAM FOR INFLIGHT USE

John Robert Bliss

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MODIFICATION OF THE
TASDA COMPUTER PROGRAM
FOR INFLIGHT USE

by

John Robert Bliss

Thesis Advisor: D. E. Harrison, Jr.

September 1973

Modification of the
TASDA Computer Program
for Inflight Use

by

John Robert Bliss
Commander, United States Navy
B.S., Miami University, 1958

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1973

[illegible]

ABSTRACT

TASDA, acronym for Tactical Airborne Sonar Decision Aid, is a computer simulation designed to select optimum sonobuoy pattern spacings given environmental parameters and submarine mode of operation. The program was designed to operate in a Tactical Support Center for briefing of flight crew personnel. Analytical methods and statistical models are used to investigate the TASDA program with a view towards modifying it for future aircraft inflight utilization. Some improvements are made to the TASDA model which reduce program run time and core storage requirements. A modified version of the TASDA program is developed as an initial step toward an inflight model.

TABLE OF CONTENTS

TABLES

FIGURES

I. THE TASDA PROGRAM

A. INTRODUCTION

TASDA, an acronym for Tactical Airborne Sonar Decision Aid, is a computer simulation program which assists tactical coordinators of P-3, ASW aircraft, to determine optimum sonobuoy search patterns and buoy spacing for varying oceanographic conditions, submarine types, and submarine mode of operation.

Initial specifications for TASDA were written by Mr. M. L. Metersky of the Naval Air Development Center, Warminster, Pa., in April, 1971. The computer program was written by International Business Machines, Inc., under Navy contract numbers N-62269-72-C042 of January 11, 1972 and N-62264-73-C0289 of January 25, 1973.

B. NEED FOR SIMULATION

One of the most difficult tasks in antisubmarine warfare is to obtain an initial contact on a submarine. For patrol aircraft, the primary means of submarine detection is the sonobuoy, an air-dropped, expendable, passive sonar device.[1] The passive sound system of the sonobuoy relies on transmission of submarine acoustic energy in the water for contact.

---

[1]Although active-type sonobuoys are used in ASW, these are not used for initial detection of submarines.

Detection ranges for passive sonobuoys vary from less than one mile to several hundred miles. The detection range depends upon many variables, such as hydrophone depth, submarine depth, condition of detection equipment, noise frequency and intensity generated by the submarine and, above all, the prevailing seawater conditions. Because of the numerous variables and their complex dependencies, it is very difficult, if not impossible, to calculate optimum sonobuoy patterns and sonobuoy spacing within these patterns, through the use of deterministic models. However, it is possible to calculate a probabilistic solution to the problem of optimum sonobuoy deployment through the use of computer simulation of the real world environment. TASDA is a laboratory model which was written to show the feasibility of applying computer simulation techniques to the airborne ASW problem.

C. DESCRIPTION

The basic simulation approach used in TASDA is to move a submarine along a randomly selected path within a given area of ocean which has predetermined seawater characteristics. Several different sonobuoy geometries or patterns are tested in the ocean area to determine whether the submarine will be detected. After numerous simulated submarine runs, a probability of detection (Pd) is calculated for each sonobuoy geometry and each different spacing within these geometries as the ratio of the number of detections to the total number of submarine runs.

A more detailed discussion of the TASDA program operation follows.

The user chooses one of three possible types for solution:

1. selection of the sonobuoy geometries and spacings which give detection probabilities equal to or greater than an input probability;

2. selection of the pattern spacings which maximize probability of detection for each relevant geometry;

3. selection of the sonobuoy geometry and spacing which maximize area of coverage for a given minimum Pd threshold.

Once a problem type has been specified, the user must then input known or suspected submarine, aircraft, environment, and sonobuoy characteristics. These include the type of submarine, either nuclear or conventional; the type of submarine operation, either transiting or on station; the submarine speed and, if a conventional submarine, both its snorkeling and submerged speed; the minimum and maximum times for snorkeling and submerged operations for conventional submarines; the coordinates of the area in which a submarine initiates its motion, and the area to which it is transiting, or the minimum and maximum bearing angles the submarine might use in moving from its original position; the operating depth of the submarine; the minimum and maximum number of sonobuoys to be considered in a geometry; the number of buoy information processing channels which the aircraft has

available; the expected time late for the aircraft arrival on-station; the radio range of the sonobuoys; the hydrophone depth of sonobuoys; the desired probability of detection threshold (not included for probability optimization problems); the number of figures of merit (FOM) and the specific FOM values to be tested.

FOM is an acoustic parameter which describes the target, ocean, receiver, and receiver operator.

$$FOM = SL - AN - RD$$

In the above formula, SL represents the target's acoustic intensity, AN represents the ambient sea noise and RD represents recognition differential. RD, which takes into account receiver and display accuracy, and operator ability, is defined as the minimum signal-to-noise ratio which enables an operator to detect a target on his receiver or display equipment 50 percent of the time when a target is present. All the terms in the above formula are measured in decibels.

The FOM, combined with the sound propagation loss over the distance from the submarine to a sonobuoy, determines whether a detection will be made.

A sound energy propagation loss profile, in one mile increments out to 204 miles, from a propagation loss program which is run prior to initiating TASDA, is part of the input. Although the propagation loss profile dominates the calculation of meaningful real world numbers, it will not be treated

in detail here, because the generation of propagation loss profiles is beyond the scope of this investigation.

A file of sonobuoy geometries is a necessary input to the program, but, because of the complex format of the geometry data, this file must be designed and input prior to program initiation. This file includes an identification label for each geometry, the type of submarine operations for which the pattern is devised, the number of buoys in the pattern, the buoy subsets to be processed together (if the aircraft can not process all buoys at once), the monitoring time of each subset of buoys, a spacing parameter specified in dimensionless units internal to the program, called grid units, which determines the relative size of each pattern, and x and y position of each buoy in a rectangular coordinate system measured in grid units, spacing ratio increment used to increase or decrease pattern size, and, finally, a geometry radius, which is the distance from the center of the pattern to the furthest buoy.[2]

The initial step toward the solution of any of the three problem types is the selection of a buoy geometry which satisfies the requirement for the minimum and maximum number of sonobuoys allowed, and which is compatible with the assumed submarine mode of operation. Once a geometry is selected, the minimum and maximum plausible spacings between

---

[2]The relationship of grid units to final pattern spacing is discussed in detail in Appendix A.

the sonobuoys in the pattern calculated. An initial sonobuoy spacing is calculated which is based on the size of the operating area input by the user and a spacing parameter which is a predetermined value calculated for each geometry. An initial spacing increment is computed in the same manner. This increment is used to vary the size of the spacing parameter which is used to calculate the spacing for successive probability of detection computations for the selected geometry.[3] Next, depending on the type of problem selected by the user, a submarine is simulated to move through the designated area; its starting position and direction of movement are calculated at random within prespecified limits. A pseudo random number generator is used to assign appropriate randomness to the original position and direction. Movement of the submarine is done in time steps. A position for the submarine is calculated for each step, and at each position a calculation is made to determine whether each buoy in the geometry is in contact with the submarine. This determination is based on (a) the range from the submarine to the sonobuoy, (b) the sound propagation loss profile, and (c) the FOM adjusted on a random basis for each sonobuoy and for each run of the submarine. If the propagation loss for the computed sonobuoy to submarine distance is less than the adjusted FOM, a detection is credited for that run. When the first detection of a given run occurs, the FOM is

---

[3]Sonobuoy pattern spacing and geometry location in the operating area is discussed in detail in Appendix A.

increased slightly to adjust for an alerted operator. Following each run for a given spacing, the probability of detection is calculated. The sonobuoy pattern spacing is changed after each set of runs until the probability of submarine detection specified by the user is exceeded or a maximum probability of detection has been achieved. Other parameters of interest are also computed. These include the Pd for simultaneous detection on three or more buoys, the mean contact times for single, double and multiple detections and the mean time to the first submarine detection.

A non-linear program search routine is used to calculate the spacing increment to be used in determining buoy spacing and to decide when an optimal probability of detection has been found.[4] The program tests each sonobuoy geometry in turn until all pertinent geometries have been exhausted and all relevant information has been printed.

TASDA was written specifically for two purposes. First as a laboratory model, it was to demonstrate the feasibility of simulation as an effective tool to optimize sonobuoy geometries and spacings under given operating conditions. Secondly, it was designed to operate in a Tactical Support Center where results from the model could be used to brief flight crews prior to an ASW mission.

---

[4]The non-linear program search routine is discussed in detail in Appendix B.

To ensure that flight crew briefings are complete and that most ASW situations are covered, the following four features are incorporated in TASDA.

First, there is a provision to handle up to six given figures of merit (FOM) for any single problem. This allows the tactical coordinator to be briefed on the best sonobuoy geometries and spacings available for a given selection of probable ambient noise conditions which will be encountered on a mission.

Secondly, it has a sonobuoy monitoring cycle capability. This permits realistic problem solutions in those situations where the number of sonobuoys in a pattern exceed the number of receivers and processing channels available in the aircraft. Presently, sonobuoy geometries used in TASDA have as many as 16 buoys, while aircraft processing capabilities vary from four to 16 channels, depending on the model and its modifications.

Third, it has the capacity to generate random snorkel cycles for conventional submarines to be used in problem solutions where target submarines are known to be diesel powered.

Fourth, there are two separate subroutines for calculating probabilities of detection. One is used to compute a probability of detection for a given geometry and its spacing. This is utilized when computer run time is critical and only single probability of detection results are required. The second subroutine requires more computer time for Pd

calculations, but it also computes probabilities for two and three simultaneous detections by different sonobuoys, mean time to first detection and mean holding times for single and multiple sonobuoy detections.

Although most of these features are necessary to accomplish the mission for which TASDA was designed, they make the program long, complex, and slow in terms of run time.

II. OBJECTIVES

With the advent of the P-3C Update system, the computer in the aircraft is capable of handling a simulation in addition to its many other functions. However, TASDA, in its present form, is too lengthy and too slow to be of any value to the Update system.

The primary objective of this thesis was to reduce the size of the current TASDA program and to decrease the program run time; so it would be feasible for the real time applications in the P3C 'Update' system. Secondary objectives included:

1. performing preliminary sensitivity analysis on variables in the program
2. performing statistical analysis to determine the number of submarine runs required for given confidence limits and allowable error percentages
3. reviewing the program to ensure that sound simulation procedures were used.

There were several sub-models in the TASDA program which appeared superfluous for an on-board the aircraft, real time system. Also, there were algorithms used in the TASDA model which seemed inefficient and which needed to be modified to achieve a more effective simulation. The following submodels and algorithms were investigated in light of the primary objective.

A. MODIFICATION OF PROBABILITY SUBROUTINES

The TASDA program contained two subroutines which calculated the probability of detection. The best features of each subroutine have been combined into a single subroutine. These subroutines had a substantial amount of similar logic. The subroutine labeled, PROB, was the more complex of the two. It calculated not only the probability of a single detection, but also the probability of two and three or more simultaneous detections, the mean time to first detection, and the mean time in contact for one, two, and three or more simultaneous detections.

The algorithm moved the submarine in steps from its original position until it reached a precalculated maximum distance. At each step a new submarine position was computed together with its distance from each sonobuoy in the pattern. Sonobuoy to submarine distance was then used as an index in the propagation loss profile vector to retrieve the appropriate propagation loss value. This figure was then compared to the input FOM modified for sonobuoy and run deviations to establish whether detection has occurred. Once detection was gained, the FOM was increased by two decibels to account for an alerted operator, the various time values were recorded, the number of detections was increased by one and the stepping process was continued until the submarine had moved its maximum distance.

In the PROBF subroutine, the maximum possible FOM was compared with the propagation loss profile vector to secure the maximum distance at which detection could possibly occur. For each run of the submarine, the closest point of approach (CPA) to each sonobuoy was calculated. If CPA was greater than the maximum detection range for a sonobuoy, that sonobuoy was deleted from further calculations for the run. If CPA was less than maximum detection range, the limits along the submarine track for possible detection were computed. The submarine was then stepped along only that portion of its track where possible detection could occur, and submarine to sonobuoy distances were computed for those buoys, which had a chance of detection. Once detection was made, the run was terminated and the number of detections was increased by one. PROBF was much faster because only sonobuoys which had a chance of detection were tested and solely for the relevant portion of the submarine track. Depending on the FOM and propagation loss profile, large savings in program run time could be achieved. However, subroutine PROB returned much more information which was useful to the tactical coordinator in decision making. By combining PROB and PROBF into a single subroutine, redundant logic was eliminated and program run time was decreased while the total information output, currently provided by PROB, was retained.

B. DELETION OF MULTIPLE FOM PROVISION

TASDA presently has provisions to handle up to six given figures-of-merit for any one problem solution, but, for an

aircraft system, only one FOM is required. Multiple FOM calculations are necessary for a shore based TASDA system to ensure that the tactical coordinator of an ASW flight crew is briefed for best sonobuoy geometry selection under a selection of the ambient noise conditions which are most likely to be encountered during a mission. The incorporation of the six figures-of-merit option in the program requires extensive logic and computer memory space, but it was efficient in terms of run time. The alternative would have been to run the complete program as many times as there were desired solutions for different FOM. However, in a real time situation, where the aircraft can obtain an on-the-spot ambient noise reading and compute the FOM, there is need for only a single FOM capability within the program.

C. DELETION OF SONOBUOY MONITORING CYCLE PROVISION

The logic which determined monitoring cycles for sonobuoy pattern sub-fields was removed from the program. Present day P-3 aircraft are able to process and display from four to 16 channels of sonobuoy information at one time, depending on the model of aircraft. Because TASDA was designed for tactical support centers which cater to all models of the P-3, it was necessary to put logic in the program which developed sonobuoy monitoring cycles for those aircraft which are not equipped to receive information from all the sonobuoys operating at a given instant. However, the P3C has a 16 channel processing capability which is adequate for continuous

monitoring of all the sonobuoys in patterns which are currently used in airborne ASW. Therefore, the monitoring cycle logic for the in-flight program was not required, and its elimination would decrease overall program size.

D. DELETION OF DIESEL SUBMARINE PROVISION

It appeared reasonable to delete from TASDA the extra programming logic required for calculating random snorkel cycles of diesel submarines. The majority of antisubmarine warfare operations in the future will be in the detection, localization, and tracking of nuclear submarines. Although conventional submarines remain a threat today, the major sea powers are rapidly converting to all nuclear powered undersea forces. The requirement for economy in program size in an in-flight system and the future reduction in the diesel submarine threat made this change to TASDA appear sound.

E. INCREASED TIME INCREMENT FOR SUBMARINE MOVEMENT

It appeared that program run time could be reduced, if the time increment used to compute the incremental distance of submarine movement in the Pd submodel was made a function of submarine speed. The Pd submodel, used in TASDA, was tested for submarine detection by comparing a FOM to the acoustic propagation loss over the distance from a sonobuoy to the submarine. This test was made after each increment of submarine movement. The distance increment was calculated by multiplying the input submarine speed by a fixed time interval of either two or four minutes. The time interval

selected was based on the time required for the submarine to traverse the operating area: if less than two hours, the two minute interval was used; otherwise, the four minute interval was used. When submarine speeds of less than 15 knots were used, the resulting distance increment would always be less than one mile. However, the propagation loss profile values were figured for one mile increments. Therefore, many redundant FOM and propagation loss comparisons would be made whenever submarine speed was less than 15 knots, because the propagation loss value remained the same until the sonobuoy to submarine distance changed by one mile. The small time increments in TASDA were necessary when sonobuoy monitoring cycles were utilized to ensure sonobuoys were tested for possible submarine detection while the buoys were being monitored. However, as discussed in Section C, there is no requirement for the monitoring cycle algorithm in an airborne TASDA model.

F. INCREASED PROPAGATION LOSS VECTOR INDEX INCREMENT

It appeared possible to decrease program run time by increasing the index increment size used to retrieve propagation loss values from the propagation loss vector. In the PROBF subroutine, a decrementing index was used to retrieve values from the propagation loss vector; each value was then tested to determine if it was less than the sonobuoy FOM. The propagation loss index corresponded to the distance for which the propagation loss value was figured. When the

first propagation loss value was found, which was less than the sonobuoy FOM, its index value was used as the sonobuoy-to-submarine distance in an algorithm which calculated points on the submarine track where first and final detection could occur. By increasing the index increment size, fewer FOM propagation loss comparisons would be required to obtain the desired sonobuoy-to-submarine distance and program run time was reduced.

G. REPLACEMENT OF RANDOM NUMBER GENERATOR SUBROUTINE

Program run time could be decreased by replacing the random number generator subroutine call instructions with the random number generator itself. The pseudo random number generator used in TASDA was written as a subroutine which contained two Fortran instructions. This subroutine was called from ten separate instructions within the program, but could be exercised over 400,000 times during the period of a complete program execution. The time to execute subroutine calls was eliminated at the expense of extra required memory space.

H. SENSITIVITY ANALYSIS OF SONOBUOY AND RUN DEVIATION

In the real world, there are differences in the sensitivity of sonobuoys of the same type and manufacture, and distinct points within a specified area of ocean will have unequal propagation loss profiles. These variations were introduced into the program by altering the FOM. Random normal deviations were produced by an algorithm which utilized pseudo random

numbers and a given standard deviation. A distinct FOM value was assigned to each sonobuoy for each submarine run. This value was the sum of a base FOM plus a random normal deviation for the sonobuoy plus another random normal deviation for the submarine run. No testing had been carried out to determine whether the added randomness significantly changed program results. Preliminary sensitivity analysis was conducted to determine whether significant differences existed in program results when sonobuoy and run deviations were deleted.

I. RANDOM NUMBER GENERATOR SEED REINITIALIZATION

The seed value for the random number generator should be reinitialized to its original value prior to each set of submarine runs in the Pd submodel. The TASDA program was designed to provide ASW flight crews with the best information on sonobuoy geometries and the spacing of buoys within each geometry for given sea water conditions and submarine operations. Since geometries were compared to one another, they should have been tested under the same conditions; that is, variables outside the geometry type and sonobuoy spacing should have been constant for every geometry. In the TASDA model, this was not the situation, as the random number generator was initialized with a specific seed value at the start of the program, then continues to generate different random numbers throughout the entire solution process. This procedure produces different submarine tracks for each

geometry and for each spacing within the geometry. This would have little effect on program results if a sufficiently large number of submarine runs were generated for each spacing of each geometry; i.e., as the number of runs approaches infinity, results would approach their true values. Because of time limitations, relatively few runs were generated for each spacing; so, to reduce the variation between sets of runs, the seed value should be reinitialized. The method described will cause identical submarine tracks to be generated for each spacing and geometry, thus a truer comparison can be made. This will not eliminate the possibility that the final solution is incorrect, compared to the true solution for an infinite number of runs; but it does ensure the best solution for those submarine tracks which are generated.

J. STATISTICAL ANALYSIS FOR SOLUTION ACCURACY

The concept underlying any simulation is the generation of probabilistic solutions to problems which can not be solved by other means. On the assumption that a good random number generator is used and that the assumptions concerning real world inputs are correct, the accuracy of simulation solutions is dependent on the number of program iterations, or trials, employed. Essentially, as the number of trials approaches infinity, the simulation solution approaches the true solution. Unfortunately, computer run time limits the number of trials for any simulation; so statistical methods

must be utilized to determine the number of trials required for any desired solution accuracy. Submarine runs through a specified area were the trials in TASDA. However, a set number of runs were used in this simulation without regard to solution accuracy. There was a need to statistically determine the number of submarine runs required for prespecified solution accuracy under the conditions imposed by a specific set of statistical assumptions.

III. METHODOLOGY

A. PROGRAM PREPARATION

A version of the TASDA III program was obtained in December, 1972 from the Naval Air Development Center. The program was identical to the one being exercised at that time at NADC on a UNIVAC 1230 computer. However, two subroutines were missing which had been on a separate magnetic tape at NADC. The subroutines were the pseudo random number generator and one which read the geometry files into the program. These two subroutines were rewritten by the author and placed in the TASDA program. The program was then exercised in its many different modes to ensure there were no syntax or logic errors which would cause the program to fault. There was no way to compare the results of the program run on the IBM 360 computer at the Naval Postgraduate School with those obtained at NADC, because the pseudo random number generators would produce different numbers. Therefore, it was assumed that the results obtained on the IBM 360 computer were mechanically correct; that is, they differed from the results obtained at NADC only because the random numbers were unequal and not due to program changes. Results obtained from the basically unaltered TASDA program were saved as control numbers to ensure later alterations to the program would not change the basic answers.

B. MODIFICATION OF THE PROBABILITY SUBROUTINE

The basic concept behind modifying the two probability subroutines was to reduce the number of times sonobuoys were tested to determine if they were in contact with the submarine. In the subroutine, PROB, the submarine was moved along its entire track in small distance increments. For each increment, every sonobuoy in the pattern being used was tested for possible contact with the submarine. This process was relatively slow in that, for each test, the sonobuoy to submarine distance had to be figured which required the use of a square root routine. The subroutine, PROBF, had an algorithm which, for each sonobuoy, determined whether contact was possible and, if possible, the points along the submarine's track where first and last submarine contact could be made. The basic change to the program was to insert the PROBF algorithm, just described, into the PROB subroutine prior to the point where the submarine was stepped along its track. Then, for each track segment, it was first determined if each sonobuoy could possibly make contact, prior to testing if, in fact, it was in contact. The following modifications to subroutine PROB were made:[5]

1. Following initialization of variables, the PROBF algorithm was inserted into the subroutine. This algorithm

---

[5]For actual program changes, see Appendix C, and the listings of subroutine PROB.

was slightly modified in order to calculate those points along the sub's track where the first and last possible contacts by any sonobuoy could occur. These two points delineated the portion of the track where the sonobuoys would be tested for contact with the submarine.

2. A flag was set for each sonobuoy that could not make contact during the current run of the submarine.

3. The initial submarine coordinates were moved to coincide with the point where the first possible detection could be made. This eliminated any needless detection testing prior to possible contact.

4. The total distance the submarine had moved along its track was calculated to compare with the distance where last contact could be held by each buoy.

5. A short algorithm was placed at the beginning of the sonobuoy testing loop to ensure that only those sonobuoys that were in detection range of the submarine at its latest position would be tested to be in contact. Three tests were made. Had the submarine moved past the point on its track where the indexed sonobuoy could last hold contact? Could the indexed sonobuoy make contact with the submarine during the remaining portion of the submarine run? Had the submarine reached the point along its track where the indexed sonobuoy could acquire first contact? If contact could not be held by the indexed sonobuoy, the program would loop back to check the next sonobuoy. This reduced to a

minimum the relatively slow tests to determine whether a sonobuoy was in contact with the submarine.

C. DELETION OF UNNECESSARY PROVISIONS

The deletion of the programming logic for the provisions for multiple figures-of-merit, for sonobuoy monitoring cycles, and for diesel submarine snorkel cycles required a thorough investigation of each subroutine in the TASDA program to ensure the remaining program logic would not be altered. The TASDA program was exercised in all problem modes, with the same inputs, before and after the logic was deleted from the program. All problem solutions were identical; so, it was assumed that the remaining program was not inadvertantly changed beyond the desired deletions. To save programming time and computer run time, all three provisions were deleted at once. Detailed discussion of which logic was removed will not be set forth in this thesis, because it would be very lengthy and it is not considered significant to program understanding. However, a comparison of the program listings will show the changes that have been made.

D. INCREASED TIME INCREMENT FOR SUBMARINE MOVEMENT

The following discussion pertains to the TASDA program without the provisions for sonobuoy monitoring cycles and submarine snorkel cycles.

Two possible alternatives for increasing or changing the time increment were investigated. First, the time increment was removed, as a data statement input, and made a

user-controlled input. The program was then exercised, holding all inputs constant, except the time increment. This was increased for each program run until a significant change in results was noted. Then, the time increment, prior to the significant change, was considered the best to use as an input. However, this was done for only one submarine speed; so results could vary significantly if a faster submarine speed were used as an input. The second approach was to make the time increment a function of submarine speed. This was done in the main subroutine by setting the time increment equal to the time it takes the submarine to move one mile at its input speed. Since the propagation loss profile is in one mile increments, this assured no propagation loss value would be overlooked for any sonobuoy as the submarine was moved along its track. Results from the first method of obtaining an increased time increment indicated that significant changes to program solutions do not occur until the time increment exceeds the time it takes the submarine to move two miles at its input speed; so the formula was changed to set the time increment equal to the time it takes the submarine to move two miles at its input speed.

E. INCREASED PROPAGATION LOSS VECTOR INDEX INCREMENT

The following discussion pertains to the probability subroutine discussed in Section B of this chapter.

The algorithm in the probability subroutine, which compared a sonobuoy FOM with successive propagation losses

for decreasing distance increments to determine the distance where first contact could be made, was altered in the following manner.

The increment size was changed from a constant equal to one to a user input variable to allow for ease of testing different increment sizes and to add flexibility to the program. When increment size, other than one, was input, the algorithm would decrease the distance by that amount after each comparison where propagation loss exceeded FOM, until either distance went to a negative number or FOM was greater than or equal to the propagation loss. When distance became negative, it implied no detection possible on that sonobuoy, and the procedure was repeated for the next sonobuoy. When FOM was greater than or equal to propagation loss, distance was increased by one less than the increment, the increment was changed to one, and the process of comparing was continued until FOM was greater than propagation loss again. Thus, the distance where propagation loss first becomes less than FOM is realized. This distance is the value required by the algorithm to compute the initial and final possible contact points for a sonobuoy.

F. REPLACEMENT OF RANDOM NUMBER GENERATOR SUBROUTINE

The random number generator subroutine was deleted from the program. The calls to this subroutine were replaced by the two Fortran instruction , pseudo random number generator. Prior to instituting this change, two small programs were

written; so the run time difference for a given number of runs, in executing the instructions of the pseudo random number generator and in executing a call to a subroutine which had the identical instructions, could be compared.

G. SENSITIVITY ANALYSIS OF SONOBUOY AND RUN DEVIATIONS

A preliminary analysis of the effects to solution results in removing the random normal deviations for sonobuoys and/or for runs was conducted. The program was exercised in four different configurations while all input variables were held constant. The four configurations were:

1. sonobuoy and run deviations left in the program
2. sonobuoy deviations removed and run deviations left in the program
3. run deviations removed and sonobuoy deviations left in the program
4. both sonobuoy and run deviations removed from the program.

The results of these program runs were tabulated, comparison graphs were made of a few selected results, and statistical analysis was carried out on the probability of detection solutions for single sonobuoys and on two sonobuoys. The statistical method employed was to test the hypothesis that mean values of Pd for two different configurations were the same over several different sonobuoy spacings. It was assumed each submarine run was a Bernoulli trial; therefore, the distribution for Pd for each spacing was binomial. The

angular, or arcsin, method of variance stabilization was used, because several binomial distributions were being averaged.[6] The null hypothesis was that the mean values of the samples were equal. The alternative hypothesis was that the mean values were not equal.

H. SEED REINITIALIZATION

The following discussion pertains to the TASDA program with the provisions for diesel submarine snorkel cycles removed. Random numbers were generated for three different algorithms within subroutine PROB. These numbers were utilized to obtain the submarine's initial position and course, to calculate a sound energy deviation for each submarine run, and to calculate a sound energy deviation for each sonobuoy. Because it was desired to compare each spacing and each geometry against the identical set of submarine tracks and identical FOM values for each sonobuoy, it was necessary to use two random number generators. One supplied random numbers for computing run sound energy deviations and for calculating submarine course and initial position, while the other was used for generating the sound energy deviations for each sonobuoy. If only a single random number generator was employed, the random numbers used for calculating submarine track and run deviations would differ between geometries,

---

[6] Brownless, K.A., Statistical Theory and Methodology in Science and Engineering, pp. 113-114, John Wiley and Sons, Inc., 1960.

using unequal numbers of sonobuoys. The random number generator used in TASDA generates each successive seed value from the previous seed value; so to ensure identical random numbers between two geometries, not only do the seeds have to be reinitialized before each call on the PROB subroutine, but also the quantity of random numbers must be equal. When the number of sonobuoys between two patterns vary, the quantity of random numbers used for each submarine run during Pd calculations will not be equal. Therefore, after the first submarine run, the random numbers for computing submarine tracks, run deviations, and sonobuoy deviations would be different for the two geometries; thus, different submarine tracks and FOM values would result. By using two random number generators, the random numbers for calculating submarine tracks and run deviations could be kept apart from the effects of unequal sonobuoy numbers between different geometries.

An alternative to reinitializing the seed value for the random number generator was also investigated. Instead of reinitializing the seed value, an algorithm was placed in the PROB subroutine which stored in computer memory each submarine track and sonobuoy FOM value for the first geometry and spacing tested during program operation. These stored values were then retrieved from memory for all successive spacings and geometries during Pd calculations. This achieved the same result of comparing identical submarine track and

sonobuoy FOM values between geometries and spacings that seed reinitialization did.

I. STATISTICAL ANALYSIS

The method employed for statistical analysis for the number of submarine runs required was to assume each run was a Bernouilli trial; that is, for each run, the submarine was either detected or not detected. The outcome of each run was assumed to be an independent random variable. The above assumption was considered valid, because the submarine track is generated by the use of random numbers. Further, since the true Pd is always an unknown quantity, it was desirous to measure the probability that the sample Pd, obtained as a solution, would differ from the true Pd by a given percentage of the true Pd. This was represented by the following equation:

$$\Pr\{\left|\frac{S_n}{N} - p\right| \leq \varepsilon p\} = 1 - \alpha$$

where
S_n = total number of detections
N = total number of runs
p = the true Pd (always unknown)
ε = percent allowable error
$1-\alpha$ = probability that sample Pd varies from true Pd by less than allowable percentage of true Pd

With the above assumptions and relation, the central limit theorem was used to calculate the approximate number of trials required to assure that the program solution for Pd

is within a specified percentage of the true p (unknown) for a given confidence level.[7]

$$N = \left(\frac{K_{(1-\alpha)}}{\varepsilon} \right)^2 \frac{1-p}{p}$$

where $K_{(1-\alpha)}$ = the ± Z value or the standard normal deviations corresponding to the area under the standard normal curve represented by the confidence level $(1-\alpha)$.

This equation was not made an integral part of the TASDA program, but was investigated to determine the number of submarine runs required for various error allowances and confidence levels where true probabilities of detection were assumed.

[7]Hoel, P. G., Port, S. C., Stone, C. J., Introduction to Probability Theory, pp. 190-191, Houghton Mifflin Co., 1971.

IV. RESULTS AND CONCLUSIONS

A. FORWARD

Results relating to computer run time and computer core area are based on the use of the Fortran IV programming language, and the IBM 360/67 computer which are employed at the Naval Postgraduate School, Monterey, California. Program results obtained from other computer facilities utilizing a different computer system could vary quite markedly from those reported here. Run time results do not include compile or high speed printer time; however, such results can vary a few seconds from actual run time because of computer interrupts.

Three separate versions of the TASDA program are mentioned in the text. TASDA Mod III refers to that program received from the Naval Air Development Center, which remains unaltered with the exception of those changes necessary to make the program run on the IBM 360 computer used at the Naval Postgraduate School. TASDA UPDATE refers to an altered TASDA Mod III program in which the subroutines PROB and PROBF have been replaced by a single subroutine incorporating features from both of them. TASDA AC refers to the TASDA Mod III program which has been altered to be employed in an aircraft.

B. MODIFICATION OF THE PROBABILITY SUBROUTINE

After the new probability subroutine had been written and debugged, so that the TASDA update program solutions were identical to those obtained from the TASDA Mod III program, the following test runs were made on each program to compare run times.

1. Program set up included the following inputs: threshold of Pd = .85, transiting nuclear submarine, eight channels of monitoring capability, submarine speed - 15 kts., one figure of merit = 80, propagation loss profile formula utilized, area of transit equal 280 by 100 nautical miles, total operation area equal 320 by 100 nautical miles, ten submarine runs per probability calculation.

The run time for the TASDA Mod III program utilizing the PROB subroutine equaled 2 minutes 45 seconds and, utilizing PROBF, equaled 11 seconds; the run time for the TASDA UPDATE program equaled 15 seconds.

2. Program set up the same as above with the following exceptions: problem was area maximization with a holding submarine, and maximum distance equaled 150 nautical miles. The TASDA Mod III program run time utilizing the PROB subroutine equaled 3 minutes 44 seconds, while it took the TASDA UPDATE program 23 seconds to obtain the same results. The PROBF subroutine may not be used when the area maximization problem is selected. The number of computer words used for each subroutine was 1365 for PROB, 1372 for PROBF, and 1674 for the modified PROB subroutine.

The program run time, when the modified subroutine was used, was obviously much faster than when PROB was used; yet, the same results were realized. There was a small increase in run time when the modified subroutine was compared with PROBF; but this was not considered significant, because of the extra information obtained when using the modified subroutine.

Although the modified subroutine is more than 300 words longer than either PROB or PROBF, there was a net savings of 1063 words when the two subroutines were replaced by the modified one.

C. ELIMINATION OF PROVISIONS

The elimination of the provisions for the multiple FOM capability, diesel submarine snorkel cycles, and sonobuoy monitoring cycles had no significant effect on program run time. A comparison test was conducted where program set up was such that these provisions were not utilized in the TASDA Mod III program.

Set up was as follows:

nuclear transiting submarine, one FOM = 80, 16 sonobuoy monitoring capability, ten submarine runs per probability calculation. Both programs did include the new Pd subroutine to save computer run time. There was no appreciable difference in run time between the two programs. For the TASDA Mod III program, run time equaled 12.05 seconds, while the TASDA AC program ran 11.91 seconds to achieve the same

solutions. This was expected, as both programs used essentialy the same statements which asked the type of submarine and the number of monitoring channels available which were included in the unaltered program. Overall program size was reduced significantly with the removal of the above named provisions. TASDA Mod III required approximately 32,000 words of core memory, while the TASDA AC program required approximately 19,000 words of memory for a reduction of 13,000 words, a 40.5 percent reduction.[8] Therefore, it appeared worthwhile to remove the previously discussed provisions for an aircraft TASDA system.

D. INCREASED TIME INCREMENT FOR SUBMARINE MOVEMENT

1. Tests of Different Time Increments

The TASDA Mod III program used time increments of two minutes for the small step size and four minutes for the large step size. To test the effect these time increments had on program run time, the TASDA AC program was exercised with seven different sets of time increments: 2-4 minutes, 4-8 minutes, 5-10 minutes, 6-12 minutes, 7-14 minutes, 8-16 minutes, 9-18 minutes. The program set up was as follows: nuclear transiting submarine, 15 knot speed, 320 by 100 nautical mile area, optimum Pd threshold, propagation loss formula used, 100 runs per probability calculation. Since the area was large, the greatest time increment of each set was utilized by the program. The 15 knot

---

[8] Total program core memory requirement is printed out in 500 word increments, up to the next 500 words.

submarine speed was used to establish a base set of Pd figures, because the test for Pd would occur at a maximum of one mile increment change in distance from submarine to sonobuoy; so no propagation loss values would be missed. The results are listed in Table I. Program run time decreased as the time increment for step size increased. However, the run time decreased at a decreasing rate. The reduction in run time was 27.2 percent between time increments of four and eight minutes, 6.7 percent between eight and ten minutes, 4.6 percent between 12 and 14 minutes, and 1.7 percent between 16 and 18 minutes. The percentages cited can not be treated in absolute values; because, as was earlier stated, program run times which are printed out can vary a few seconds, depending on the number of computer interrupts generated during the period the program is running.

The sonobuoy spacing solution for a given geometry did not change for any of the time increments.

Detection probabilities decreased as the time increment increased, which was expected; because minimum sonobuoy to submarine distances would be bypassed, thus decreasing the chance that FOM would be tested against the propagation loss value associated with the shortest distance. Except for convergence zones, propagation loss values are normally smaller at shorter ranges. There was no discernible rate of drop off of Pd values, and, for the five geometries tested, the amount of decrease in Pd ranged from 12 percent to four percent.

TABLE I

PROGRAM SOLUTIONS FOR VARIED
TIME 'STEP' INCREMENTS

SET UP: PROPAGATION LOSS FORMULA, 15 KNOT TRANSITING SUB, FOM = 75, OPTIMUM THRESHOLD, 100 RUNS PER PROB. CALCULATION

PD1 = PROBABILITY OF SINGLE BUOY DETECTION
PD2, PD3 = PROB. OF TWO AND THREE OR MORE SIMULTANEOUS DETECT.

GEOMETRY IDENT.	TIME INCREMENT	BUOY SPACING	PD1	PD2	PD3
	4	19	.79	.26	.11
	8	19	.79	.26	.11
	10	19	.76	.24	.09
BD 16	12	19	.74	.24	.10
	14	19	.72	.24	.10
	16	19	.72	.22	.10
	18	19	.68	.22	.10
	4	14	.79	.31	.17
	8	14	.79	.31	.17
	10	14	.79	.31	.17
NCT 16	12	14	.77	.30	.17
	14	14	.77	.30	.17
	16	14	.74	.30	.16
	18	14	.73	.27	.17
	4	28	.59	.09	.02
	8	28	.59	.09	.02
	10	28	.58	.09	.02
BD 9	12	28	.57	.09	.02
	14	28	.57	.08	.02
	16	28	.56	.07	.02
	18	28	.55	.07	.02
	4	19	.70	.26	.10
	8	19	.70	.26	.09
	10	19	.69	.26	.09
NT 8	12	19	.66	.26	.09
	14	19	.64	.26	.09
	16	19	.59	.26	.08
	18	19	.58	.26	.09
	4	40	.58	.02	.01
	8	40	.57	.02	.01
	10	40	.57	.02	.01
CB 8	12	40	.55	.02	.01
	14	40	.54	.02	.01
	16	40	.53	.02	.01
	18	40	.52	.02	.01

PROGRAM RUN TIME FOR
VARIED TIME INCREMENTS

TIME STEP	RUN TIME
4	1 MIN 43 SEC
8	1 MIN 15 SEC
10	1 MIN 10 SEC
12	1 MIN 06 SEC
14	1 MIN 03 SEC
16	1 MIN 00 SEC
18	0 MIN 59 SEC

The test was run a second time, utilizing a different initial seed value for the pseudo random number generator. The results, which are listed in Table II, were very similar to those in the first test, except for a logic weakness in the search algorithm which caused erroneous solutions for one of the geometries. The search algorithm will be discussed later in this chapter.

It was concluded that program run time can be effectively reduced by increasing the time step increment. There will be a reduction in the values of Pd, but best spacings for each geometry will not change. It should be restated that these results and conclusions pertain to the TASDA AC program in which sonobuoy monitoring cycles and submarine snorkel cycles were removed. It appears evident that different results would be obtained if these features were exercised with a larger time step, because detection opportunities would be lost when a sonobuoy monitoring event or a snorkel cycle were overstepped by a large step size.

After the time increment was made a function of submarine speed, the program was exercised with three different input speeds — 5, 15, and 20 knots. The program solutions for all detection probabilities were identical, but the mean holding times varied inversely to the submarine speed. This was as expected, because the step distance increment is treated as a constant value when the step time value is made a function of submarine speed: the equation used being, step time (minutes) = (step distance in nautical miles times

TABLE II

PROGRAM SOLUTIONS FOR VARIED TIME 'STEP' INCREMENTS

SET UP: PROPAGATION LOSS FORMULA, 15 KNOT TRANSITING SUB, FOM = 75, OPTIMUM THRESH-OLD, 100 RUNS PER PROB. CALCULATION

PD1 = PROBABILITY OF SINGLE BUOY DETECTION
PD2,PD3 = PROB. OF TWO AND THREE OR MORE SIMULTANEOUS DETECT.

GEOMETRY IDENT.	TIME INCREMENT	BUOY SPACING	PD1	PD2	PD3
	4	19	.81	.15	.07
BC 16	8	19	.80	.15	.06
	10	19	.78	.14	.06
	14	19	.76	.13	.06
	4	58	.44	.02	.01
NCT 16	8	58	.44	.02	.01
	10	58	.42	.02	.01
***	14	24	.74	.06	.04
	4	28	.68	.07	.03
BC 9	8	28	.66	.07	.03
	10	28	.65	.07	.03
	14	28	.59	.06	.03
	4	19	.59	.15	.04
NT 8	8	19	.59	.15	.04
	10	19	.58	.15	.04
	14	19	.56	.15	.04
	4	40	.55	.04	.01
CB 8	8	40	.54	.02	.01
	10	40	.53	.02	.01
	14	40	.51	.02	.01

PROGRAM RUN TIME FOR VARIED TIME INCREMENTS

	TIME STEP	RUN TIME
	4	1 MIN 26 SEC
	8	1 MIN 02 SEC
	10	0 MIN 57 SEC
***	14	1 MIN 08 SEC

*** DIFFERENCE IN SPACING AND PD'S DUE TO WEAKNESS IN SEARCH ALGORITHM

60) divided by submarine speed. The above equation also explains the reason for the changing of the mean holding times. Program run times for the different speed inputs varied within two seconds of each other. Again, this result was expected, because the program operated with identical figures with the exception of time values which are only summed for mean holding time and mean time to first detection read-outs. Program run times were also within a few seconds of the run time of the unaltered TASDA AC program, where input time increment step size resulted in a two mile distance increment.

It was concluded that the TASDA AC program could be more effectively written by changing the time step increment from a fixed input value to a value which is a function of submarine speed. This would eliminate the possibility of making redundant FOM versus propagation loss comparisons when the submarine's speed is such that it would move less than one mile in a fixed time step. However, by increasing the constant step distance used in the equation, the same program run time savings are realized; and once step distance is set, program run times and Pd results will no longer vary with different input submarine speeds.

E. INCREASED PROPAGATION LOSS VECTOR INDEX INCREMENT

The TASDA AC program was used to test the new algorithm which determined at which range the submarine could first be detected by a given sonobuoy. Program initial set up was:

transiting submarine, speed 15 knots, optimum Pd, FOM = 75, 100 runs, 64 points of propagation loss data. The propagation loss data were obtained from Fleet Weather Central Naval Facility at Monterey, California. These data, listed in Table III, were calculated from actual oceanographic information and were specifically chosen, because convergent zone conditions were represented. One, three, and ten mile values were used as the index increment on successive program runs. Program run time decreased from four minutes 31 seconds for the one mile index increment, to three minutes 26 seconds for the three mile increment, and one minute 52 seconds for the ten mile increment. It should be noted that run times were considerably higher when actual propagation loss data were used rather than the propagation loss formula.

Final program solution results, which are listed in Table IV, were identical for optimum sonobuoy spacing for each geometry tested, using the three different index increments. There was one exception at the increment size of ten for geometry NT-8, but this was caused by a weakness in the search algorithm. This weakness is discussed in Section J of this chapter. Probability of detection for single and multiple sonobuoy contacts decreased as the index increment was increased. This was expected, because convergence zone detections can be lost when best propagation loss values in a convergence zone are not tested against sonobuoy FOM. This can occur any time the index increment is greater than one, and the likelihood of this occurring increases as the

TABLE III

PROPAGATION LOSS DATA

ACTUAL P.L. DATA FROM
FLEET NUMERICAL WEATHER CENTER

SET UP: SOURCE DEPTH 539 FT.
RECEIVER DEPTH 300 FT.
SEA STATE 2

DIST. FROM SRC NAUTICLE MILES	+1	+2	+3	+4	+5
0	63.3	68.1	71.1	72.1	73.1
5	74.4	76.6	80.8	88.3	83.8
10	85.4	86.5	94.7	94.0	86.8
15	83.1	83.5	86.3	88.7	93.5
20	97.1	90.8	87.9	85.6	83.7
25	82.7	82.9	84.2	86.3	88.8
30	91.0	92.2	93.2	93.9	92.8
35	90.3	88.1	86.8	86.0	85.9
40	86.0	86.3	86.5	86.6	86.8
45	87.0	87.1	87.1	87.2	86.9
50	86.7	86.4	86.3	86.2	86.2
55	86.0	85.4	84.5	84.7	82.7
60	80.5	78.8	78.4	79.0	79.2

TABLE IV

PROGRAM SOLUTIONS FOR VARIED
P.L. PROFILE INDEX INCREMENTS

SET LP : 64 POINT PROPAGATION LOSS PROFILE DATA, 15 KNOT TRANSITING SUBMARINE, FOM = 75, 100 RUNS PER PROB. CALCULATION

PD1 = PROBABILITY OF SINGLE BUOY DETECTION
PD2, PD3 = PROB. OF TWO AND THREE OR MORE SIMULTANEOUS DETECT.

GEOMETRY IDENT.	INCREMENT SIZE	BUOY SPACING	PD1	PD2	PD3
BC 16	1	19	.83	.56	.37
	3	19	.83	.54	.32
	10	19	.82	.20	.09
NTC 16	1	24	.86	.52	.34
	3	24	.86	.45	.29
	10	24	.84	.18	.07
BC 9	1	28	.76	.48	.19
	3	28	.76	.42	.16
	10	28	.70	.12	.07
NT 8	1	19	.76	.40	.17
	3	19	.76	.37	.14
***	10	91	.30	.04	.02
CB 8	1	40	,70	.31	.12
	3	40	.70	.28	.11
	10	40	.63	.11	.07

PROGRAM RUN TIME FOR
VARIED INCREMENT SIZES

INCREMENT	RUN TIME
1	4 MIN 31 SEC
3	3 MIN 26 SEC
10	1 MIN 52 SEC

*** DIFFERENCE IN SPACING AND PD'S DUE TO WEAKNESS IN SEARCH ALGORITHM

index increment size increases. Pd for single sonobuoy contacts did not differ between increment sizes of one and three, but varied from one to seven percent between increments of one and ten. However, Pd for multiple simultaneous contacts varied as much as 36 percent.

It was concluded that program run time could be decreased significantly by increasing the index increment for the propagation loss profile vector. There would be a small cost in required memory space and a loss in Pd for single and multiple simultaneous contacts.

F. REPLACEMENT OF RANDOM NUMBER GENERATOR SUBROUTINE

A small algorithm was placed in the TASDA program to count the number of times the random number generator was called for each program run. For the TASDA AC program, where the number of submarine runs per probability calculation was set to 100, the random number generator was used an average of 450,000 times per program run. This average figure was then used in the two programs, written to test run times for straight execution of the random number generator and for the execution with a subroutine call. The time for execution of only the instructions was 19 seconds, while the execution of the subroutine call plus the instructions was 47 seconds. The two FORTRAN instructions required for the random number generator were compiled into 15 machine instructions, the subroutine compiled into 51 machine instructions, and each subroutine call required five machine instructions; so, to

replace the ten random number generator subroutine calls with the random number generators, costs 49 additional memory locations.

It was concluded that run time can be decreased by not using the random number generator in a subroutine. Time savings becomes a function of the user input of the number of submarine runs used per probability calculation, and it is significant for problems where 100 runs or more are designated.

G. SENSITIVITY ANALYSIS ON SONOBUOY AND RUN DEVIATIONS

The program run set up for the statistical analysis was a nuclear, transiting submarine, 15 knot speed, one FOM = 75, threshold optimum and propagation loss formula, 250 runs per probability calculation. Program results are listed in Tables V, VI, and VII. Three different geometries were tested at a level of significance of 0.05. The null hypothesis, that the means for Pd were equal for different configurations, was rejected if the test statistic was greater than $K_{(1-\alpha)}$ or 1.96. Test statistics are listed in Table 8. There was no significant difference at 0.05 LOS for single buoy Pd for programs run with both sonobuoy and run deviations included and programs run with only run deviations included. This was true for all three geometries. There were significant differences for single buoy Pd between runs with both sonobuoy and run deviations included and runs with both deviations not included. There were mixed results among

TABLE V

SENSITIVITY ANALYSIS PROGRAM SOLUTIONS GEOMETRY ID BD-16

TRANSITING SUBMARINE
FIGURE OF MERIT = 75
250 RUNS PER PROB.CALCULATION
PROPAGATION LOSS FORMULA USED

PD1 = PROB. OF SINGLE DETECTION
PD2 = PROB. OF TWO OR MORE BUOYS IN CONTACT SIMULTANEOUSLY
PD3 = PROB. OF THREE OR MORE SIMULTANEOUS BUOY CONTACTS
HT1 = MEAN HOLDING TIME IN MIN. FOR SINGLE BUOY CONTACTS
HT2 AND HT3 = MEAN HOLDING TIMES IN MIN. FOR TWO AND THREE SIMULTANEOUS CONTACTS

		SONOBUOY SPACING IN NAUTICLE MILES							
		5	12	19	26	32	37	56	75
BUOY AND RUN DEV-IATIONS IN PROGRAM	PD1	.468	.724	.756	.664	.660	.612	.436	.283
	PD2	.336	.296	.156	.068	.056	.044	.016	.008
	PD3	.256	.148	.068	.036	.032	.016	.004	.004
	HT1	94	69	63	65	65	67	70	69
	HT2	81	67	87	136	23	104	141	162
	HT3	77	75	104	87	85	125	88	24
SONOBUOY DEVIATIONS DELETED FROM PROGRAM	PD1	.440	.732	.748	.632	.640	.612	.440	.292
	PD2	.300	.180	.084	.048	.024	.020	.004	.000
	PD3	.204	.068	.040	.020	.008	.004	.000	.000
	HT1	64	55	49	53	51	51	54	52
	HT2	61	46	80	73	80	74	51	0
	HT3	71	90	78	82	144	84	0	0
RUN DEVIATIONS DELETED FROM PROGRAM	PD1	.432	.808	.860	.768	.680	.572	.436	.264
	PD2	.392	.176	.004	.000	.000	.000	.000	.000
	PD3	.336	.000	.000	.000	.000	.000	.000	.000
	HT1	87	42	34	33	33	34	35	31
	HT2	51	16	24	0	0	0	0	0
	HT3	28	0	0	0	0	0	0	0
BUOY AND RUN DEV-IATIONS DELETED	PD1	.428		.764		.620	.548	.408	.212
	PD2	.376		.000		.000	.000	.000	.000
	PD3	.304		.000		.000	.000	.000	.000
	HT1	77		28		28	28	29	28
	HT2	44		0		0	0	0	0
	HT3	10		0		0	0	0	0

TABLE VI

SENSITIVITY ANALYSIS PROGRAM SOLUTIONS GEOMETRY ID NTC-16

TRANSITING SUBMARINE
FIGURE OF MERIT = 75
250 RUNS PER PROB.CALCULATION
PROPAGATION LOSS FORMULA USED

PD1 = PROB. OF SINGLE DETECTION
PD2 = PROB. OF TWO OR MORE BUOYS IN CONTACT SIMULTANEOUSLY
PD3 = PROB. OF THREE OR MORE SIMULTANEOUS BUOY CONTACTS
HT1 = MEAN HOLDING TIME IN MIN. FOR SINGLE BUOY CONTACTS
HT2 AND HT3 = MEAN HOLDING TIMES IN MIN. FOR TWO AND THREE SIMULTANEOUS CONTACTS

		SONOBUOY SPACING IN NAUTICLE MILES					
		5	14	24	29	43	58
BUOY AND RUN DEVIATIONS IN PROGRAM	PD1	.448	.752	.732	.688	.484	.401
	PD2	.332	.244	.096	.060	.032	.012
	PD3	.292	.104	.040	.036	.008	.004
	HT1	95	70	66	67	79	80
	HT2	77	73	138	174	122	113
	HT3	74	106	158	118	122	152
SONOBUOY DEVIATIONS DELETED FROM PROGRAM	PD1	.408	.728	.748	.668	.448	.388
	PD2	.284	.092	.044	.032	.008	.004
	PD3	.200	.052	.028	.016	.004	.000
	HT1	68	52	51	53	57	58
	HT2	63	89	110	88	83	27
	HT3	73	111	73	84	60	0
RUN DEVIATIONS DELETED FROM PROGRAM	PD1	.428	.836	.876	.812	.508	.488
	PD2	.396	.060	.000	.000	.000	.000
	PD3	.336	.000	.000	.000	.000	.000
	HT1	87	37	34	34	38	34
	HT2	58	16	0	0	0	0
	HT3	28	0	0	0	0	0
BUOY AND RUN DEVIATIONS DELETED	PD1	.404	.824	.844	.744	.480	.432
	PD2	.360	.000	.000	.000	.000	.000
	PD3	.308	.000	.000	.000	.000	.000
	HT1	77	27	28	28	28	28
	HT2	37	0	0	0	0	0
	HT3	16	0	0	0	0	0

TABLE VII

SENSITIVITY ANALYSIS PROGRAM SOLUTIONS GEOMETRY ID BC-9

TRANSITING SUBMARINE
FIGURE OF MERIT = 75
250 RUNS PER PROB.CALCULATION
PROPAGATION LOSS FORMULA USED

PD1 = PROB. OF SINGLE DETECTION
PD2 = PROB. OF TWO OR MORE BUOYS IN CONTACT SIMULTANEOUSLY
PD3 = PROB. OF THREE OR MORE SIMULTANEOUS BUOY CONTACTS
HT1 = MEAN HOLDING TIME IN MIN. FOR SINGLE BUOY CONTACTS
HT2 AND HT3 = MEAN HOLDING TIMES IN MIN. FOR TWO AND THREE SIMULTANEOUS CONTACTS

		SONOBUOY SPACING IN NAUTICLE MILES					
		5	28	52	57	85	113
BUOY AND RUN DEVIATIONS IN PROGRAM	PD1	.352	.608	.416	.378	.280	.244
	PD2	.264	.064	.008	.007	.000	.000
	PD3	.196	.028	.004	.004	.000	.000
	HT1	81	61	59	63	63	63
	HT2	70	92	124	76	0	0
	HT3	68	83	120	88	0	0
SONOBUOY DEVIATIONS DELETED FROM PROGRAM	PD1	.344	.552	.376	.372	.264	.224
	PD2	.228	.044	.004	.004	.000	.000
	PD3	.184	.016	.000	.000	.000	.000
	HT1	61	53	53	51	51	48
	HT2	62	68	76	44	0	0
	HT3	52	106	0	0	0	0
RUN DEVIATIONS DELETED FROM PROGRAM	PD1	.364	.632	.396	.368	.264	.176
	PD2	.316	.000	.000	.000	.000	.000
	PD3	.256	.000	.000	.000	.000	.000
	HT1	63	36	33	33	34	34
	HT2	40	0	0	0	0	0
	HT3	24	0	0	0	0	
BUOY AND RUN DEVIATIONS DELETED	PD1	.344	.572	.360	.324	.236	.168
	PD2	.300	.000	.000	.000	.000	.000
	PD3	.248	.000	.000	.000	.000	.000
	HT1	59	29	28	29	30	29
	HT2	36	0	0	0	0	0
	HT3	14	0	0	0	0	0

TABLE VIII

TEST STATISTICS FOR SENSITIVITY ANALYSIS

DIFFERENCE IN MEANS BETWEEN TWO SETS
OF BINOMIAL DISTRIBUTIONS USING
VARIANCE STABILIZING TRANSFORMATION

$$\text{TEST STATISTIC} = \left| 1/M \sum_{j=1}^{M} (\phi_{1j} - \phi_{2j}) \sqrt{M/2} \right|$$

M = NUMBER OF BUOY SPACINGS PER GEOMETRY

$\phi = 2\sqrt{N}\,\text{ARCSIN}(\sqrt{\hat{P}})$

N = NUMBER OF RUNS PER PROB. CALCULATION

$\hat{P}$ = COMPUTED PD FOR SINGLE BUOY SPACING

CONFIGURATION	GEOMETRIES					
	BD 16		NTC 16		BD 9	
BUOY AND RUN DEVIATIONS IN VERSUS	PD1 TS	PD2 TS	PD1 TS	PD2 TS	PD1 TS	PD2 TS
BUOY DEVIATIONS DELETED	0.545	5.033	1.099	5.190	1.405	1.214
RUN DEVIATIONS DELETED	2.147	10.73	4.786	9.628	0.869	3.393
BUOY AND RUN DEVIATIONS OUT	2.266	11.28	2.479	12.23	2.773	3.552

the three geometries when the testing was between runs with both deviations in the program and only buoy deviations in the program. For two geometries, single buoy Pds were significantly different, but for the third geometry there was no significant difference. There were significant differences for Pds of simultaneous contact on two or more sonobuoys, when buoy or run deviations were deleted from the program.

Figures 1, 2, 3, and 4 illustrate examples of (a) no significant difference for single buoy Pd when sonobuoy deviations were deleted; (b) no significant difference for single buoy Pd when run deviations were deleted; (c) significant difference for single buoy Pd when run deviations were deleted; (d) significant difference for multiple buoy Pd when buoy deviations were deleted. These graphs begin on page 57.

Program run times for the four configurations were as follows: buoy and run deviations in, two minutes 54 seconds; only buoy deviations, two minutes 27 seconds; only run deviations, one minute 53 seconds; buoy and run deviations deleted, one minute 16 seconds.

It was concluded that, on this preliminary analysis, program run time could be reduced approximately 35 percent by removing sonobuoy uncertainties from the program, and this would not significantly change the results of single sonobuoy Pd. However, results for multiple simultaneous detections

FIGURE 1

BUOY AND RUN DEVIATIONS VS. RUN DEVIATIONS

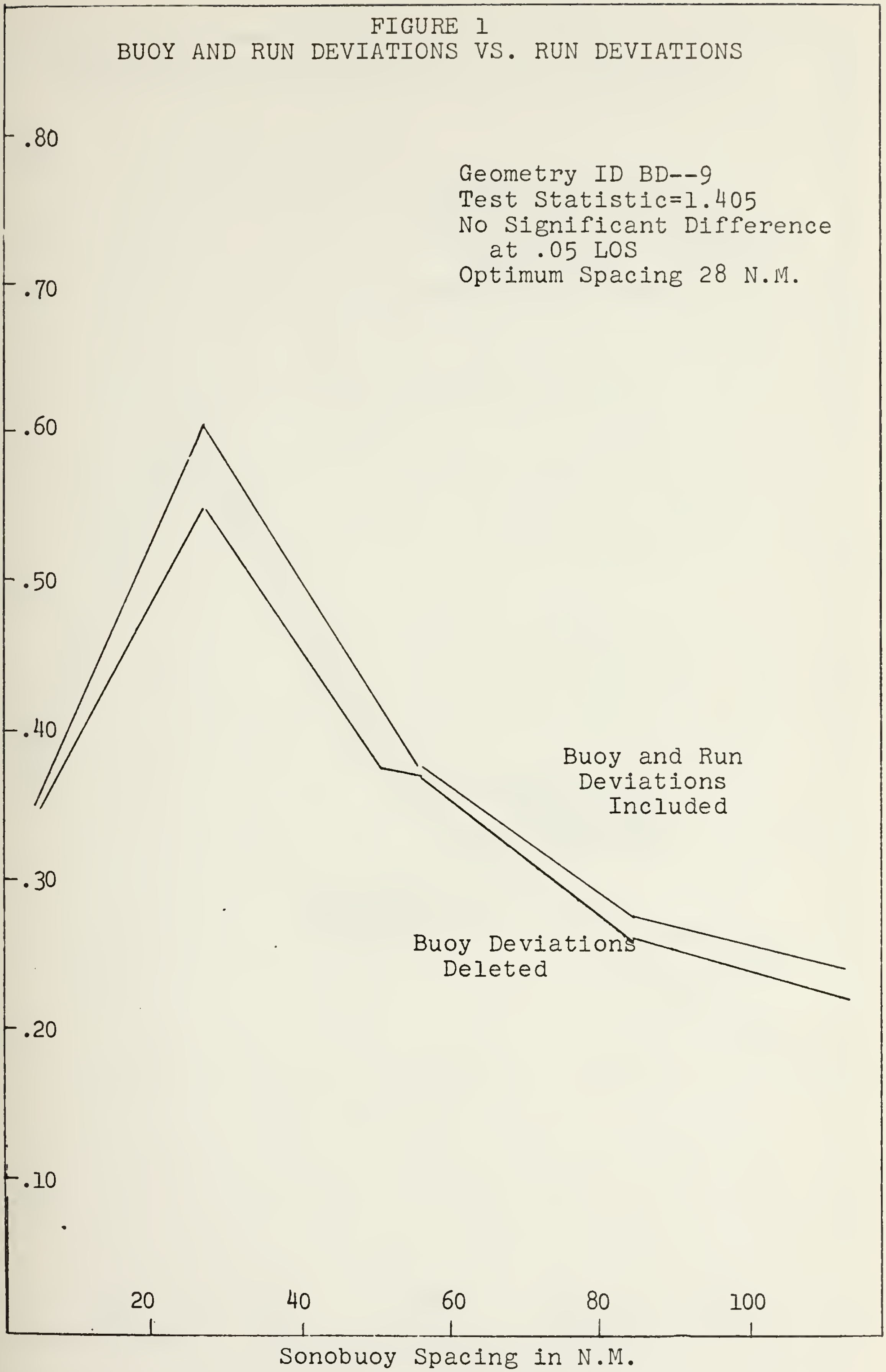
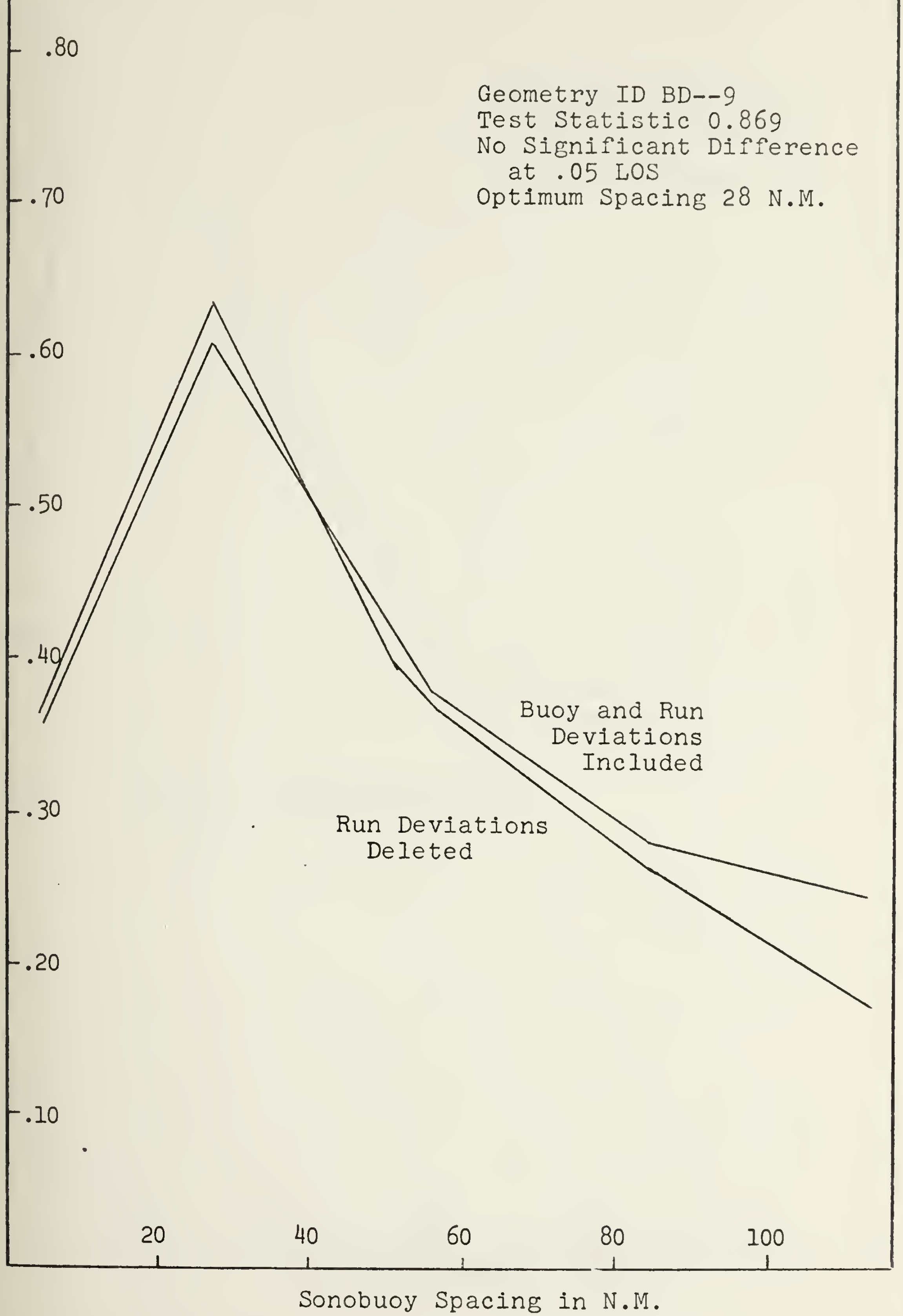


FIGURE 2
BUOY AND RUN DEVIATIONS VS. BUOY DEVIATIONS



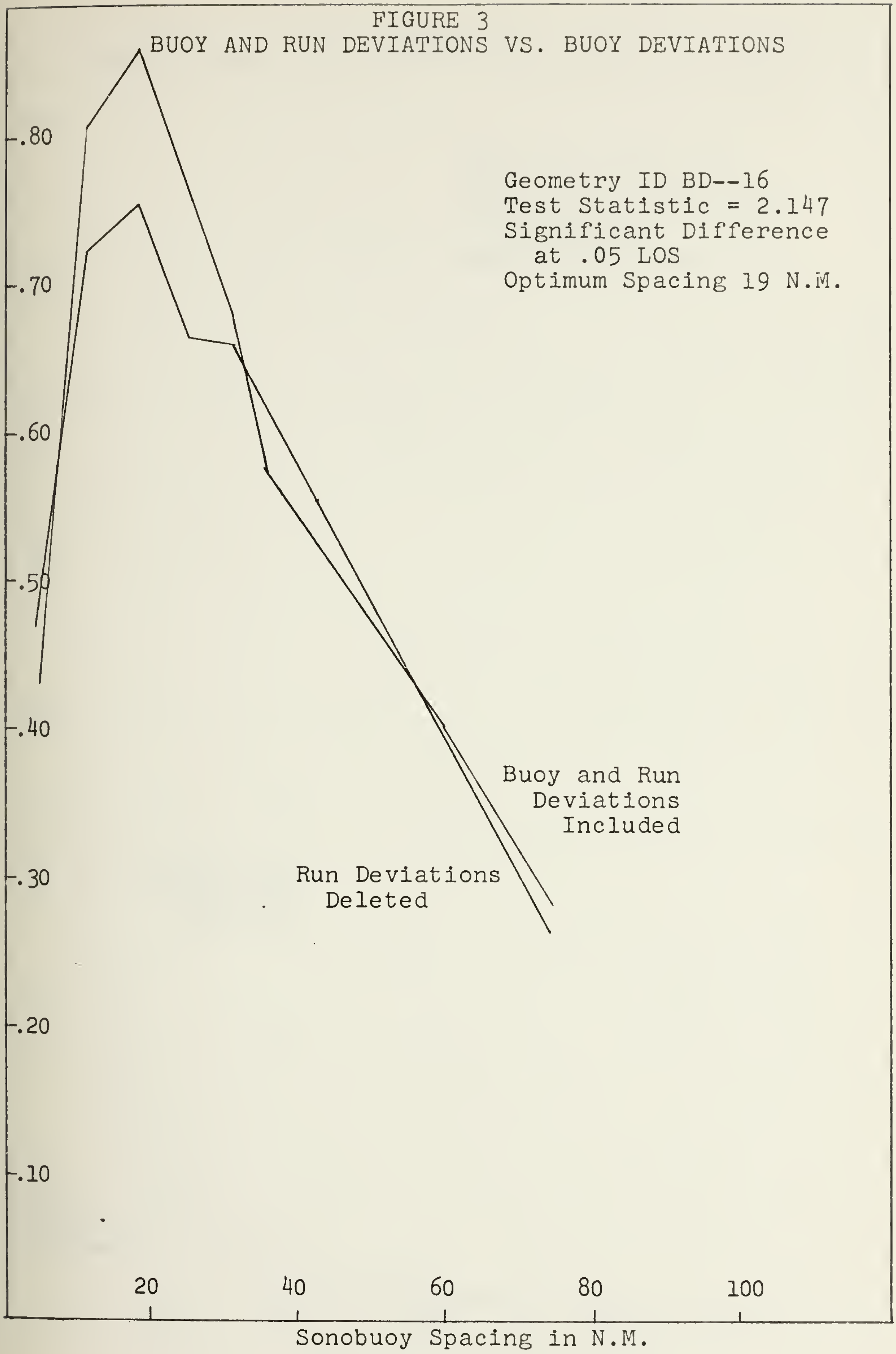
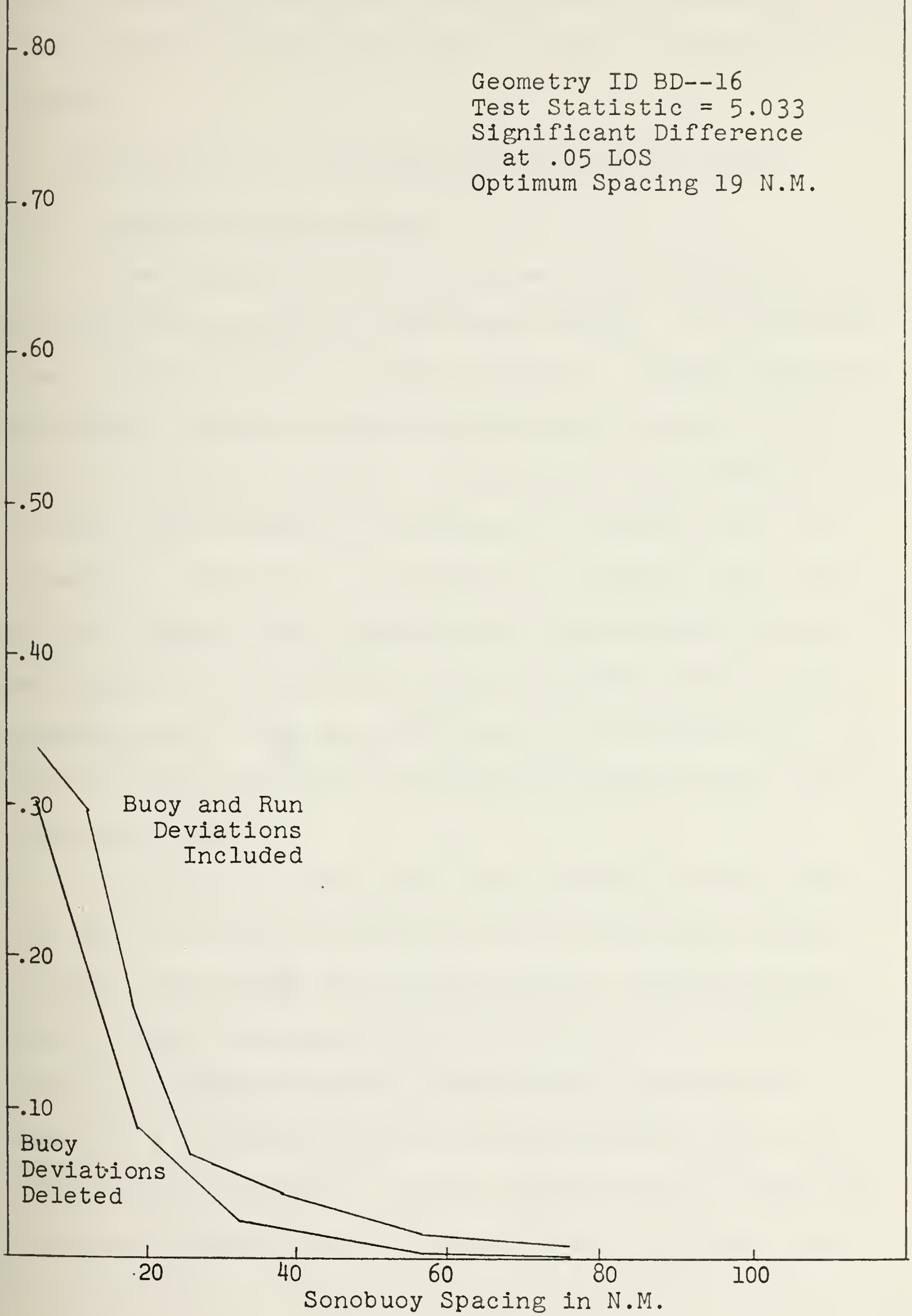


FIGURE 3
BUOY AND RUN DEVIATIONS VS. BUOY DEVIATIONS
.80
.70
.60
.50
.40
.30
.20
.10
Geometry ID BD--16
Test Statistic = 2.147
Significant Difference
at .05 LOS
Optimum Spacing 19 N.M.
Buoy and Run
Deviations
Included
Run Deviations
Deleted
20
40
60
80
100
Sonobuoy Spacing in N.M.

FIGURE 4
BUOY AND RUN DEVIATIONS VS. RUN DEVIATION MULTIPLE Pd



would be significantly altered. Removal of run uncertainties, or both buoy and run uncertainties, would significantly change program results, although run time economies could be realized.

H. REINITIALIZATION OF RANDOM NUMBER GENERATOR SEED

1. Seed Reinitialization

Two complete TASDA AC program runs were made -- one with seed reinitialization and one without -- to test run time, program size and solution results. Program setup was as follows: nuclear transiting submarine, speed 15 knots, FOM = 75, threshold optimum, propagation loss formula and 100 runs per probability calculation. Results are listed in Table 9. There was no increase in program size, because only the place in the program where the seed was initialized was changed. Seed initialization was moved from the main program where it was done only once to the subroutine, TRSH, where the seed would be reinitialized prior to each call on subroutine, PROB.

Program run times were not compared because the program with seed reinitialization tested five more sonobuoy spacings for Pd than the program without reinitialization. Using the same statistical test as described in Chapter III., Section H, program solution results were significantly different at a 95 percent level of confidence for two of the five geometries tested. It was concluded the 100 runs per probability calculation were insufficient to ensure there

TABLE IX

R.N. SEED REINITIALIZATION
VERSUS NO REINITIALIZATION

SETUP: NUCLEAR TRANSITING SUB,
15 KNOT SPEED, FOM = 75,
PROPAGATION LOSS FORMULA,
THRESHOLD OPTIMUM, 100 RUNS
PER PROB. CALCULATION

PD = PROBABILITY OF
DETECTION
SPA = SONOBUOY SPACING IN N.M.
REIN = PROGRAM WITH
REINITIALIZATION

GEOMETRY	SEED									
		SPA	5	*19	32	37	56	71		
BD 16	REIN	PD	.53	.79	.67	.62	.53	.36		
	NO	PD	.48	.74	.58	.58	.54	.32		
		SPA	5	*14	24	29	43	58		
NTC 16	REIN	PD	.52	.79	.68	.68	.58	.48		
	NO	PD	.38	.74	.70	.72	.63	.47		
		SPA	5	*28	52	57	85	113		
BD 9	REIN	PD	.44	.59	.47	.45	.36	.31		
	NO	PD	.35	.52	.40	.37	.29	.24		
		SPA	5	*12	*19	23	33	43	67	91
NT8	REIN	PD	.46	.68	.70	.53	.50	.36	.20	.17
	NO	PD	.34	.65	.61	.55	.54	.45	.32	.24
		SPA	5	*40	75	80	100			
CE 8	REIN	PD	.40	.57	.33	.27	.20			
	NO	PD	.37	.53	.32	.24	.20			

* INDICATES SONOBUOY SPACING WHICH IS
OPTIMAL SOLUTION FOR GEOMETRY

would be no significant differences between program results for runs with and without seed reinitialization.

2. Storage of Submarine Track and FOM Values

One TASDA AC program run was made, using the identical program set up which was utilized for the seed reinitialization program. Program results were the same as the seed reinitialization program. This was expected, because identical inputs and random number values were used. The only difference in the programs was that one recalculated track and FOM values for Pd calculations, while the other program stored the original values and then retrieved them for each Pd calculation. The stored value program run time was 48 seconds and the reinitialized seed program run time was 66 seconds. The program which utilized the stored track and FOM values required 2000 additional words of core memory. There was a requirement for 20 extra words of computer memory for each submarine run used in Pd calculations. The 20 locations included 16 for FOM values for each sonobuoy, two for the X and Y coordinates of the submarine's original position and two for sine and cosine values of the submarine's course. It was concluded that a substantial amount of run time could be saved by storing submarine track values and FOM values for each sonobuoy, but this would be at the expense of using additional core memory.

I. STATISTICAL ANALYSIS

The equation discussed in Chapter III, Section J, $N = \left(\frac{K_{(1-\alpha)}}{\varepsilon}\right)^2 \frac{1-p}{p}$, was used to determine several values for

the number of submarine runs required to obtain statistical accuracy for given confidence levels and allowable percentages of error. Since the true Pd is never known, the most conservative measure for calculating N was used. By setting p equal to .5, the maximum value of $\frac{1-p}{p}$ was obtained. If the value, 1-p, becomes greater than p, the criterion for p would be changed; i.e., p would be set equal to probability of no detection; so that value $\frac{1-p}{p}$ would remain less than one. The solutions for four sets of values for $1-\alpha$ and ε were then computed. These were as follows:

$1-\alpha = .95$ $\quad K_{(1-\alpha)} = 1.96$

$\varepsilon = .05$ $\quad$ Result $N = 1537$

$1-\alpha = .95$ $\quad K_{(1-\alpha)} = 1.96$

$\varepsilon = .10$ $\quad$ Result $N = 376$

$1-\alpha = .90$ $\quad K_{(1-\alpha)} = 1.645$

$\varepsilon = .10$ $\quad$ Result $N = 270$

$1-\alpha = .90$ $\quad K_{(1-\alpha)} = 1.645$

$\varepsilon = .15$ $\quad$ Result $N = 120$

The above results were the most conservative, no matter what the true value of p. Stated in another way, if in fact the true value for p was equal to .80 and a ten percent error were allowed, then for 270 submarine runs used to determine a Pd, one could be 90 percent confident that the solution would be between .72 and .88. However, the user never knows the true Pd, but only the sample Pd from the

program solution; so, using the sample result, the following could be concluded:

Let the sample solution equal .75. It was known for 270 runs that, at a 90 percent confidence level, .75 is within ten percent of the true Pd or $p - .10 \times p \leq .75 \leq p + .10p$. Solving for p, it is determined that the true value for p would lie within the interval from .675 to .825. Since the 270 runs were based on a p equal to .50, and the worst case was p equal .67, the error percent could be readjusted by replugging the equation with the values $270 = (\frac{1.645}{\varepsilon})^2 \times \frac{.33}{.67}$ or $\varepsilon = .071$, which implies for 270 runs the true p lies in the interval .695 to .80 for the stated confidence level of .90.

It should be observed that the number of runs in the model used, is more sensitive to changes in the percent of allowable error than to the confidence level. The reason for this was the percent allowable error values were normally chosen in a region of 10^{-1}, while the z values were on a scale of 10^{0}, thus the squared value of ε has a magnitude greater change than the squared value of $K_{(1-\alpha)}$.

J. THE SEARCH ALGORITHM

The search algorithm used in TASDA, as described in Appendix II, appeared to have a weakness in its logic. After a Pd has been calculated for the first sonobuoy spacing, the spacing is always increased for the next Pd calculation, and will continue to be increased for successive Pd calculations,

until either Pd decreases or the spacing limit is reached. If the Pd decreases, for the second calculation, then the spacing is reduced to less than that of the original spacing, and subsequent spacings will be made smaller, until Pd decreases or other stopping criteria are met. During actual program operation, if the second Pd calculation was greater than the first, and the third Pd calculation was smaller than the second, search stop criteria would be met; and the second sonobuoy spacing would be selected as the optimum spacing for the current geometry. Thus, there was no Pd calculation made for a spacing which was less than the original spacing. On two occasions, during program execution, a smaller spacing than the original one would have yielded a much higher Pd, if it had been tested; but a search stop criterion was met first. These results were confirmed by forcing a smaller spacing on the search algorithm with identical program setup.

If real world Pd versus sonobuoy spacing was a unimodal function, the above mentioned weakness of the search algorithm could still be displayed because of the relatively small number of submarine runs used for each Pd calculation. Therefore, although the search algorithm was designed to locate an optimum Pd and to start with a best guess spacing, this model can be trapped into finding a suboptimal Pd and sonobuoy spacing; because, under certain conditions, testing of spacing less than the original spacing is not accomplished.

V. RECOMMENDATIONS

A. REPLACE PROB AND PROBF

The two subroutines, PROB and PROBF, should be replaced in the current TASDA MOD III program by the modified PROB subroutine. The savings in time and core memory for the full information program far overshadow the small loss in program run time by the elimination of PROBF, which only yields limited information.

B. ELIMINATION OF PROVISIONS

For a TASDA program designed for real time in flight operation, the provisions for sonobuoy monitoring cycles and multiple FOM capability should be eliminated from the program to reduce program size. Further investigation is necessary to determine the need for the snorkel cycle option, because of the many third power countries in the world which can pose a conventional submarine threat.

C. INCREASED TIME INCREMENT

The time increment in the current TASDA MOD III program should be increased to ten or 12 minutes to take advantage of program run time savings. However, if short snorkel cycles or sonobuoy monitoring cycles are going to be used for a given program run, the time increment should be smaller than the minimum cycle time; so that cycles are not overstepped. For the aircraft version of TASDA, where sonobuoy monitoring

cycles and snorkel cycles are deleted from the program, the time increment should be made a function of submarine speed. However, further investigation is needed to determine the best step distance increment to use as a multiplier in the suggested equation.

D. PROPAGATION LOSS VECTOR INDEX INCREMENT

The propagation loss vector index increment should be increased in the current TASDA program and in any subsequent flying version. The time savings are sufficient to make the decrease in Pd values worthwhile. More investigation in the area is required to determine the amount the increment should be increased for the optimum trade-off between Pd loss and decreased program run time. However, index increment values of three or four would be safe where oceanographic conditions are not expected to include very small convergence zones.

E. REPLACEMENT OF RANDOM NUMBER GENERATOR SUBROUTINE

The random number generator should be deleted as a subroutine, and incorporated into TASDA as an algorithm at each position where the calls to the random number generator subroutine now exist. The saving in program run time is large enough to justify the added memory space required.

F. SENSITIVITY ANALYSIS ON SONOBUOY AND RUN DEVIATIONS

Although a time savings can be realized without significantly changing program results by deleting the sonobuoy

uncertainties, there is a need for further investigation in this area prior to taking such action. The investigation completed was preliminary and the standard deviation value input for the buoys was neither questioned nor altered; so, without further testing, it appears inadvisable to remove this element from the program.

G. SEED REINITIALIZATION

The random number generator seed should be reinitialized prior to each call on the subroutine PROB or the PROB subroutine should be rewritten to include storage of submarine track and FOM values. For seed reinitialization, there is no additional memory space required and little run time will be added to the TASDA MOD III or TASDA AC programs. For track and FOM storage, substantial run time savings can be realized if core storage is available. Either method will ensure the best program solutions are obtained for the submarine tracks generated.

H. STATISTICAL ANALYSIS

Users of the TASDA program should be made aware of the consequences of using too few submarine runs per probability calculation in order to save computer run time. Knowledge of the confidence level and percent error realized for a given number of submarine runs will assist the user in determining the number of runs to input for program operation.

I. THE SEARCH ALGORITHM

There is a need to correct the weakness in the search algorithm. Too often, the conditions were met which cause the algorithm to terminate prior to searching a spacing less than the original spacing when the optimum spacing was in the unsearched region.

APPENDIX A

SONOBUOY SPACING AND GEOMETRY LOCATION IN TASDA

Sonobuoy geometries are predetermined patterns which are input to the TASDA program in a specified manner. Presently, the program is limited to 16 sonobuoys for any one geometry. Before describing how ultimate spacing is obtained for any one geometry, several word definitions are necessary.

Grid: A cartesian (rectangular) coordinate system having dimensionless units with its origin at the point (0,0). Each sonobuoy position in a geometry is designated by an X and a Y coordinate value in grid units which positively locates the sonobuoy relative to the origin.

Spacing grid parameter: A predetermined value for each geometry which is a geometry input. This value is used as one of the factors in obtaining an initial spacing ratio. The spacing grid parameter is based on a nominal real world spacing of a given sonobuoy geometry and an average size search area. Two spacing grid parameters can be used for a given geometry. If the geometry is used for detection of both transiting and holding submarines, then two values would be placed in the geometry input data. Program words for these parameters are: SPGRT, spacing grid parameter transiting; SPGRH, spacing grid parameter holding.

Spacing grid increment parameter: A predetermined value for each geometry based on the value of the spacing grid

parameter. This value is used to calculate the initial increment for changing the spacing ratio when the program is using the search routine to obtain an optimal sonobuoy spacing. Two spacing grid increment parameters are input with the geometry data whenever the geometry is used for detection of both transiting and holding submarines. Program words for these parameters are: SPIGRT, spacing grid increment parameter transiting; SPIGRH, spacing grid increment parameter holding.

Spacing ratio: This value is somewhat a misnomer, since it is the product of two values and is used as the multiplier of grid units to convert them to nautical miles. The spacing ratio is initially computed by multiplying the spacing grid parameter by the approximate radius of the operating area. It may be considered the ratio of the geometry measured in nautical miles to the geometry measured in grid units. The initial value of the spacing ratio is incrementally changed by the spacing increment during the program, in order to achieve a spacing within the sonobuoy geometry which yields either an optimum probability of detection or a probability of detection which exceeds a threshold input by the user. The program word for spacing ratio is SP(i). This, being a vector of six words, in order to accommodate up to six different figures of merit.

Maximum spacing ratio: (SPMAX) This value is the maximum spacing ratio allowed for a given geometry. It is determined by dividing the maximum radio range in nautical miles by the geometry radius in grid units.

Minimum spacing ratio: (SPMIN) This value is the minimum allowable spacing ratio for a given geometry. It is determined by dividing a spacing granularity factor measured in nautical miles by the spacing in grid units between the first and second sonobuoy of the selected geometry.

The basic idea behind calculating the spacing for sonobuoy geometries is to obtain a spacing which yields an optimal probability of submarine detection in a minimum amount of program run time. To accomplish the above task, a good first guess of spacing is necessary in order to minimize the number of spacings to be tested for a given geometry. This is done by adjusting what is normally considered the best buoy spacing for the geometry. This adjustment is made based on the size of the operating area which is an input by the user. For example, if the nominal sonobuoy spacing of a particular pattern is 30 nautical miles when given a large search area, it is logical to assume that, given a confined search area, a smaller spacing between buoys would produce better results.

In actual program operation, the initial spacing ratio is determined as follows:

For transiting submarine problems, the user inputs eight coordinate positions in nautical miles, based on a cartesian coordinate system. Four coordinates labeled XS1, XS2, YS1, YS2, determine the starting rectangle for the submarine and four coordinates determine the finishing rectangle for the

submarine. The program calculates the greatest X distance and the greatest Y distance for the input operating area. The approximate radius of the operating area is then determined to be 0.5 times the maximum of the X and Y distances. For the frontal coverage problem, the approximate radius is 0.5 times the frontal coverage distance. For holding submarines, approximate area radius is defined as 0.5 times the largest side of the submarine's initial starting rectangle, plus the distance the submarine will move prior to the arrival of the aircraft on station, plus the maximum distance the submarine can move. This maximum distance is a user controlled input parameter to the program.

The approximate area radius, labeled CDX in the program, represents area size. This is multiplied by the spacing grid parameter, representing nominal buoy spacing for a given geometry, to produce the initial spacing ratio. The initial calculated spacing ratio is then used as a multiplier of X and Y buoy coordinates in grid units to convert the coordinate to nautical mile units. The equations below are listed to indicate how grid units are transformed to nautical miles.

$$\text{SP(nautical miles)} = \text{CDX (nautical miles)} \times \text{SPGRT (dimensionless)}$$

Equation for the derivation of spacing ratio by definition,

$$\text{SP (nautical miles)} = \frac{\text{(X,Y coordinate positions)(naut. mile}}{\text{(X,Y coordinate positions) (grid unit}}$$

or

X,Y coordinate position (nautical miles) =

SP (nautical miles) x (X,Y coordinate positions)(grid units)

The program assumes that the coordinate systems origins for the sonobuoy geometry and the submarine operating area are the same. To ensure that the sonobuoy geometry is located at the center of the operating area, operating area coordinates input by the user must be arranged in such a manner that the geometric center of the input area coincides with the 0,0 position or origin. This avoids additional program logic to compute area center and, either relocate all buoy positions to correspond to it, or relocate area boundary coordinates to have the origin at the geometric center.

APPENDIX B

DOCUMENTATION OF SEARCH ALGORITHM

To ensure there was no misunderstanding, Appendix B is copied verbatum from Sections 9.0 and 10.0 of the TASDA III functional specification.[8]

9.0 THRESHOLD-OPTIMIZATION (TRSH)

This subroutine finds a spacing-ratio for which the corresponding optimizing variable(s) either exceeds the threshold or is sufficiently close to the optimum value. Given a spacing ratio SP and a set of buoy coordinates (KSB(I), KYB(I)) (I=1,2, ..., NB) expressed in grid units, the buoy coordinates in nautical miles are (SP*KXB(I), SP*KYB(I)). The values of spacing-ratio to be tested must lie in the interval $SPMIN \leq SP \leq SPMAX$. SPMIN and SPMAX were computed in geometry selection.

A search algorithm is used to find the desired spacing ratio. This algorithm computes the optimizing variable(s) for a sequence of spacing ratios SP_0, SP_1, ..., SP_n. The next spacing ratio SP_{n+1} is determined from values of $SP_i (1 \leq i \leq n)$ and from optimizing variable(s) values computed at these points.

---

[8] Naval Air Development Center, Functional Specification, TASDA III, Program for Optimal Geometric Selection, pp. 47-55, Cole, S.N., International Business Machines Co., Inc., Contract N62269-72-C-0402, 1971.

If the variable to be optimized is the default variable, the speeded up probability calculations are used; otherwise, the complete set of probability calculations are done. The probability subroutines compute probabilities for several figures-of-merit simultaneously; the time required for computing several probabilities is only slightly greater than the time required for computing one probability. Thus a considerable savings can be realized if the search algorithm is performed in parallel as much as possible for the input figures-of-merit.

The set of figures-of-merit is partitioned into subsets; in the course of the search algorithm the same sequence of spacing ratios applies to all figure-of-merit in a subset. If the search algorithm specified different values of SP_{n+1} for different figures-of-merit in a subset, or if the search algorithm terminates for some but not all figures-of-merit in a subset, the subset must be split into two or more finer subsets.

In solving the area maximization problem this subroutine is used to find threshold or optimum optimizing variable(s), where the submarine traverses an area specified by the input coverage dimension. This subroutine is used repeatedly until the maximum acceptable coverage dimension is found. Thus at the subroutine entry point the set of figures-of-merit may already have been partitioned due to differences in values of coverage dimension or in spacing-ratio initial guess.

The subroutine inputs include the number, NFMS, of figure-of-merit subsets, the indexes of the figures-of-merit in each subset, the initial guesses SP(1), ..., SP(NFMS) of spacing ratios for each of the subsets, and the initial spacing-ratio increments (SPI(1), ..., SPI(NFMS). The search algorithm is expressed in terms of state-transitions of the states of the figure-of-merit subsets. In order to define the states the following notation is adopted P(x) denote a vector $P_1(x)$, ..., $P_{NFM}(x)$ of the optimizing variable(s) computed at spacing ratio x for the NFM figures-of-merit in a subset. "$P(x) \leq P(y)$" expresses that "$P_i(x) \leq P_i(y)$ for i=1, ..., NFM". The relations "$P(x) < P(y)$", "$P(x) > P(y)$", and "$P(x) \geq P(y)$" are similarly defined. SP and SPI denote values of spacing-ratio and spacing-ratio increment respectively. SP is positive, and SPI may be positive or negative.

State 0 - This is the initial state for all subsets. It denotes that no probability calculations have been performed.

State 1 - Probability values P(SP) for a single spacing-ratio have been obtained.

State 2 - All optimizing variable(s) computed thus far have increased monotonically with increasing or decreasing spacing-ratio. The highest two vectors satisfy the relation $P(SP-SPI) \leq P(SP)$.

State 3 - An approximation to the optimum spacing-ratio has been made to within SPI. Three vectors are available satisfying the relation $P(SP-SPI) \leq P(SP) \geq P(SP+SPI)$.

State 4 - The first of two steps in improving the approximation to the optimum spacing has been performed. Three vectors are available satisfying the relation $P(SP-SPI) \leq P(SP) \geq P(SP+2*SPI)$. The 2nd step is the computation at SP+SPI.

State 5 - All optimizing variables calculated thus far have increased monotonically with increasing or decreasing spacing-ratio, and the most recent spacing-ratio is very close to SPMIN or SPMAX. $P(SP) \leq P(SP+2SPI)$, and either $SP+3*SPI>SPMAX$ or $SP3*SPI<SPMIN$.

State 6 - All optimizing variables calculated thus far have increased monotonically with increasing or decreasing spacing-ratio, and the distance between the most recent spacing-ratio and one of the limits (SPMIN or SPMAX) was between $\frac{SPI}{2}$ and SPI. The value of SPI was accordingly changed such that SP+SPI equals the near limit. The vector P(SP) remains available.

State 7 - This state is used by the FORTRAN version, but is not needed to define the algorithm.

State 8 - The optimizing variables P(SP) exceed the threshold.

State 9 - A stopping criterion has been met. The optimizing variables P(SP) do not exceed the threshold but are considered optimal.

Two stopping criteria are applied. The spacing-ratio is cut down in size after reaching states 3, 5, and 6. A flag denoted by LIM is set to record that state 5 or 6 has been reached. The granularity criterion is simply the test of whether $|SPI| \leq SPMIN$.

In state 3 a local maximum has been found. For this state an additional criterion for stopping, called the optimum criterion, is whether the interpolated maximum optimizing variable exceeds P(SP) by a significant amount; the interpolated maximum is the maximum of the parabola through the points (SP-SPI, P(SP-SPI)), (SP,P(SP)), (SP+SPI,P(SP+SPI)).

Upon entry to the subroutine, set the LIM flag to OFF and assign the state 0 for all subsets; the subsets are indexed 1, 2, ..., NFMS. Store the input seed for the random number generator into ISEED(1). Starting with subset number 1 execute the search algorithm until its state is 8 or 9. Do the same for sets 2, ..., NFMS. Note that during this process the value of NFMS increases in general due to subset refinement.

The processing for subset M sometimes requires initialization if LPT equals 3 (the area maximization problem). The coverage dimension for subset M is not necessarily equal to the

coverage dimension for subset M-1; the coordinates of the starting rectangle (and for transiting motion the ending rectangle) must be redefined if the area problem is still not solved for subset M. The value (LAS(M)) of the area status for subset M is an input to this subroutine and indicates whether or not the problem is solved. If LAS(M) is less than 5, then set

XS1=XS1=YF1=YF2=-CDD(M)

XS2-XF2=CDD(M)

YS1=YS2=CDD(M)+DF

for transiting submarines, and

XS1=YS1=-CDD(M)+DF+DISMAX

XS2=YS2=CDD(M)-DF-DISMAX

for holding submarines; CDD(M) is the coverage dimension for subset M; DF is computed during initial processing; and DISMAX is a control input.

Each iteration of the search algorithm is defined by the following steps. The iteration is repeated for subset M until it reaches state 8 or 9. Some of the tests cause subset refinement when opposite results occur for at least two figures-of-merit in the subset. When refinement occurs, NFMS is incremented by 1, and the newest subset has index NFMS. The details of the refinement subroutine are desribed later. NFM denotes the number of figures-of-merit in subset M and LS(M) denotes the state of subset M.

- If LIM(M) is ON or if LS(M)=3, test the granularity criterion.

  If $|SPI(M)| \leq SPMIN$, set LS(M)=9

- If LS(M)=3, test the optimum criterion (via the refinement subroutine). Let

$$D_i = P_i(SP-SPI)$$
$$E_i = P_i(SP) \qquad i=1, \ldots, NFM$$
$$F_i = P_i(SP+SPI)$$

  If $(D_i - F_i)^2 \leq PSDIF(2E_i - D_i - F_i)$, the criterion is satisfied; PSDIF is one of the built-in inputs.

  If the criterion was satisfied for set M, set LS(M)=9, and test for refinement; if refinement occurred, set LS(NFMS)=3. If the criterion was not satisfied and refinement occurred, set LS(NFMS)=9.

- If LS(M)=0, test whether SP is within limits.

  If SP>SPMAX, set SP=SPMAX. If SP<SPMIN, set SP=SPMIN. In either case, store the random number generator seed, JJN, into ISEED(M).

- If LS(M)=1, test whether SP+SPI is within limits. If not, invert the sign of SPI and test again. If SP+SPI is still not within limits, reduce SPI by half and test again. Continue cutting SPI in half until $SPMIN \leq SP+SPI \leq SPMAX$ is satisfied.

- If LS(M)=2, test whether SP+SPI is within limits. If not, either SP+SPI>SPMAX or SP+SPI<SPMIN. Assume the first case; the rule for the lower limit is symmetric.

If $SP+\frac{SPI}{2}>SPMAX$, redefine SPI to be $-\frac{1}{2}$ times it current value and set LS(M)=5. Otherwise, if $\frac{SP+SPI}{2} \leq SPMAX$, set SPI=SPMAX-SP and LS(M)=6.

o If LS(M)$\leq$6 compute probabilities. Collect NFM figures-of-merit into an array using the indexes in the Mth figure-of-merit subset. Let

$$XSP = \begin{cases} SP & \text{if } LSM=0 \\ SP+SPI & \text{if } 1 \leq LSM \leq 6 \end{cases}$$

$$\left.\begin{aligned} XB(K) &= XSP*KXB(K) \\ YB(K) &= XSP*KYB(K) \end{aligned}\right\} \quad K=1,\ \ldots,\ NB$$

The figures-of-merit, buoy coordinates and random number generator seed are inputs to the probability subroutines. If the optimizing variable is the default variable, set JJNSAV=JJN and call the speeded up probability subroutine to calculate and save the value of AK1 for each figure-of-merit. Otherwise, use the full blown probability subroutine to calculate probabilities and save values AK1, AK2, AK3, TDM, HTM1, HTM2, and HTM3 for each of the figures-of-merit.

o Compare P(XSP) against the threshold (via the refinement subroutine). If the optimizing variables in set M exceed the threshold, set LS(M)=8, SP(M)=XSP and ISEED(M)= JJNSAV. If no refinement occurred, this iteration of the search algorithm is terminated. Otherwise, it continues for set M1 where M1=NFMS. If the optimizing variables do

not exceed the threshold, set M1=M. If refinement occurred, set LS(NFMS)=8, SP(NFMS)=XSP and ISEED(NFMS)=JJNSAV.

- If LS(M1)=0, set LS(M1)=1
- If 1≤LS(M1)≤6, compare P(SP+SPI) with P(SP) (via the refinement subroutine). Update LS, SP, and SPI according to the following table for set M1. The updated values of SP and SPI are expressed in terms of values of SP and SPI at the beginning of this step. If refinement occurred, the results of the comparison for subset NFMS is opposite the result for subset M1, update LS, SP, and SPI for subset NFMS using the opposite side of the table.

10.0 <u>REFINEMENT</u> (REFN)

A test performed on a subset of figures-of-merit may produce opposite results for two or more elements of the subset. If so, it is necessary to refine the figure-of-merit partitioning by removing one or more figures-of-merit from the given subset and creating a new subset. The tests are two-valued: (1) "does the optimizing variable value exceed a threshold?" (2) "does the optimizing variable value at one spacing exceed the optimizing variable value at another spacing?" (3) given the 2 adjacent optimizing variable values, is the center value sufficiently close to optimal?"

TABLE X

THRESHOLD-OPTIMIZATION STATE UPDATES

CURRENT STATE	P(SP+SPI)>P(SP) *			P(SP+SPI)≤P(SP)		
	LS	SP	SPI	LS	SP	SPI
1	2	SP+SPI	SPI	2	SP	-SPI
2	2	SP+SPI	SPI	3	SP	$\frac{SPI}{2}$
3	3	SP+SPI	$\frac{SPI}{2}$	4	SP	-SPI
4	3	SP+SPI	$-\frac{SPI}{2}$	3	SP	$-\frac{SPI}{2}$
5	3	SP+SPI	$-\frac{SPI}{2}$	2	SP	-SPI
6	2	SP+SPI	SPI	2	SP	-SPI

The inputs to this subroutine are the figure-of-merit subset, the number, NFMS, of figure-of-merit subsets, an indication of which test to perform, indication of which variable(s) are being optimized, and sufficient data to perform the tests. The tests mentioned above are described in detail in the threshold-optimum section.

Since there is a choice of several variables or combinations of variables to optimize, the subroutine must determine which one(s) and store their values for use in the tests. If two variables are to be optimized they must be weighted and combined prior to storing their values.

The figure-of-merit indexes in the given subset are I_1, I_2, ..., I_{NFM}. Perform the test for figure-of-merit I_1; the result of this test defines the result for the given subset, and I_1 automatically remains in the subset. Repeat the test for I_2, ..., I_{NFM}. If the result for I_j is the same as the result for I_1, I_j remains in the subset; otherwise I_j is entered into the new subset numbered NFMS. Maintain counts: the number, (N1), of figures-of-merit remaining in the current set and the number NA of figures-of-merit in subset NFMS.

After testing all of the figures-of-merit in the given subset the count NA indicates whether or not a new subset was created. If NA=0, the execution of this subroutine is terminated. Otherwise the following pertinent threshold-optimization data defined for the given subset must be copied for the new subset.

SP	-	the current spacing-ratio
SPI	-	the current spacing-ratio increment
LIM	-	an indication of whether state 5 or 6 has been reached
ISEED	-	the current stored value of the random number generator seed.
Area	-	maximization data must also be copied
SP0	-	the best spacing for coverage yielding sub-threshold values
SP1	-	a spacing for coverage yielding threshold values
CD	-	the approximation of the optimal coverage dimension for the current geometry
CD1	-	the best coverage dimension found for previous geometries
CDI	-	the current increment of coverage dimension
CDD	-	the coverage dimension being tested
LAS	-	the area-maximization state
LFA	-	an indication of whether an area limit was reached
IGEA	-	the index of the geometry yielding the best coverage

APPENDIX C

CHANGES TO PROGRAM

There were many changes to the TASDA Mod III program in order to speed up program run time and/or to reduce program size. Significant changes to the program, except for removal of the provisions for snorkel submarine cycles, sonobuoy monitoring cycles and multiple FOM, are listed in this appendix. Following the subroutine where the change occurs, is listed the program to which it belongs.

A. PROB SUBROUTINE TASDA MOD III

In order to integrate the subroutines PROB and PROBF, the following changes were made to subroutine PROB.

The two words, DSMALL and DLARGE, were dimensioned as 16 word vectors each. These vectors are set to a value equal to the distance along the submarine's track where first and final possible submarine detection can be acquired for each sonobuoy.

The DIS = DISMAX statement immediately following the 28 CONTINUE statement was deleted, because DIS is set to a different value later in the subroutine. After the statement, BS = SIN(THPX), the short algorithm to compute the run deviation was inserted in the program. This was moved forward in the subroutine to obtain values which are necessary in the PROBF portion of the subroutine. Two words, DMAX and DMIN, were added to the program and initialized to

worst values for later comparison testing. These words are finally set to the values which indicate the distance along the submarine's track where initial contact can be obtained by a buoy in the geometry, DMIN, and where final contact can be held by the last possible buoy which can gain detection, DMAX. A program loop is then initiated to compute for each sonobuoy the values for DSMALL and DLARGE. The algorithm to compute sonobuoy deviations was inserted at the beginning of the loop, followed by the computation of FOM for the currently indexed buoy.

The two words, XDIS and YDIS, were introduced to eliminate the repeated calculation of the statements XS-XB(K) and YS-YB(K), which occur in PROBF. DISTQ was introduced to eliminate extra calculations of the value,

(BS * XDIS - BX * YDIS) * * 2.

The loop for several FOM was eliminated, because this portion of the program was set up to set track distance limits only once for each buoy per submarine run. When the multiple FOM option is being used, the largest FOM must be input first, because the calculated distance along the submarine's track where buoy to sub range is tested, then includes the distances generated for lesser figures-of-merit.

The calculation of ANFM was changed to include the two decibel increment which is used for an alerted operator. This was to ensure the track length, along which buoy to submarine distance is tested, would be calculated for best FOM conditions.

The same algorithm to compute DSMALL and DLARGE was used as was programmed in PROBF, but these values are now stored in a vector to be used later in the subroutine.

The GOTO 95 statement indicates no detection is possible for the currently indexed buoy. In that case, the DLARGE value for the buoy is set to a negative one, indicating the buoy should not be tested for submarine detection later in the subroutine.

The two statements, IF(DMAX.LT.DLARGE(K))DMAX=DLARGE(K) and IF(DMIN.GT.DSMALL(K))DMIN=DSMALL(K), set the limits along the submarine's track where buoy to submarine distance is tested. Outside these limits, there is no possibility of detecting the submarine. This concluded the PROBF integration into the subroutine.

The statement, DIS = DMAX - DMIN, was used to set the distance over which buoys would be tested to be in contact with the submarine. This distance is checked to be positive; if it is negative, this indicates no possible detections on the current submarine run; so the program returns to the start of the next run.

The original PROB subroutine, from the 18 CONTINUE statement on, was then used with only minor changes.

The words, DELTAD and DELDSN, were added. These are used to increment DMIN in an algorithm to turn buoys on and off. DMIN, which is originally set to the distance along the submarine's track where initial detection can occur, is compared to DSMALL and DLARGE for each buoy at each increment

along the submarine's track. If DMIN is less than DSMALL, it indicates the submarine has not reached the detection range of that buoy; so no testing is done and the next buoy is checked. If DMIN is greater than DLARGE, it indicates the submarine has passed beyond range of the buoy for this run. DLARGE then is set to negative one; so the buoy will no longer be tested during the current submarine run. This algorithm was put in the program to eliminate the relatively slow function of testing whether a buoy is in contact with the submarine when no detection probability exists. The statements, XS=XS+BCxDMIN and YS=YS+BSxDMIN, were put into the subroutine to move the initial submarine position to that point along its track where detection is first possible.

The words, ISTART and IFINISH, were used as the loop control variables for the number of segments loop. This was done to coordinate the submarine run time and its position along its track with the same time and position generated by the subroutine before modification. This was necessary to ensure that the modifications to the subroutine did not alter the final solution results. Results were different when buoy monitoring cycles and submarine snorkel cycles were utilized in program runs.

B. BLOCK DATA (TASDA AC)

Two new common areas were dimensioned: COMMON/B/ and COMMON/C/. COMMON/B/ contains six words which are used in connection with the generation of random numbers. JCOUNT

is used as a tally to count the number of times the random number generator is used during the course of a program run. This is not essential to the program and should be removed, if the program is used operationally. KKK is the constant value, 99947, which is utilized as a multiplier in the random number generator of the IBM 360/67 computer. MYSEED and NBSEED are dimensioned as original seed values for the random number generator. They are set in the DATA statement to 1333567 and 356019, respectively. JJJ is used to store the current seed value which changes for each new random number. NBMAX is employed in an algorithm which ensures the random numbers used for sonobuoy decibel deviations are the same from geometry to geometry. It stores the value of the number of sonobuoys in the geometry which has the greatest number of buoys. COMMON/C/ has one word, NSTEP. NSTEP is used as the index increment value for the propagation loss profile vector in the algorithm which determines that distance where a sonobuoy can gain first contact on the submarine.

C. MAIN (TASDA AC)

Three read statements were added or altered. This was done to give the programmer flexibility when exercising the program. NRUNS was added to the first read statement. This enables the programmer to read in the number of submarine runs desired per probability calculation. The statement READ(5,998) STEPL, STEPB was added when the effect of the time step on program run time was being tested. This is no

longer needed in the program, because the time step was made a function of submarine speed.

The statement READ(5,999)NSTEP was added to allow the programmer to control the index increment for the propagation loss profile vector.

The two statements, STEP=STEPL and IF(RUNT.GT.5.0)STEP=STEPB, were replaced by the statement, STEP=120.0/VEL, to make STEP a function of submarine speed.

D. PROB SUBROUTINE (TASDA AC)

This subroutine is similar to the modified PROB subroutine discussed in Section A of this appendix. After the removal of provisions for snorkel cycles, sonobuoy monitoring cycles, and multiple figures of merit, the following algorithms were added. A short five instruction routine was inserted just prior to calculating the deviation values for sonobuoys. This algorithm advances the current seed value; so the buoy deviations generated for each buoy in the current geometry will be the same as those generated for buoys in other geometries. The geometries are retrieved from the geometry file in order, according to their number of sonobuoys. The geometries with the most buoys are retrieved first. NBMAX is set to the value representing the number of buoys in the first geometry retrieved in subroutine GEOM. The difference between the number of sonobuoys in the current geometry and the number of buoys in the first geometry is calculated. If the difference is equal to zero, no action is taken. The

number of random numbers generated for the current geometry will be the same as the first geometry, because the seed was reinitialized prior to the call on PROB. If there is a difference in the number of buoys for the two geometries, the value of this difference is multiplied by 12. The product represents the additional number of times the random number generator would be called if the current geometry had as many buoys as the largest geometry. The seed for the random number generator is then multiplied by the constant value, KKK, the number of times equal to the just found product. This advances the seed to a value consistent with the value used in the largest geometry.

The second algorithm introduced into the PROB subroutine allowed a variable index increment to be designated for the propagation loss profile vector. This algorithm begins with the statement, NSUM=NSTEP. NSTEP is the user-controlled input variable which designates the increment size. NSUB replaces NSTEP within the algorithm, because its value is changed. LSUB is set to NSUB minus one. It is used to reset the distance value to one less than it was prior to being decremented by the NSUB value when the FOM first exceeds the propagation loss value. If NSTEP is originally set to one, LSUB is set to a negative value to indicate that further testing at an NSTEP value of one is not necessary after the FOM first exceeds the propagation loss value. The propagation loss profile index, NR, is then set to the number of propagation loss values which have been input to the program.

Next, the current figure of merit, ANFM, is tested to be greater than or equal to the propagation loss vector value, PLC(NR). If this is false, NR is decremented by NSUB, and the test is made again. This continues until the test is true or NR becomes negative. If a successful test is made, the value of NSUB is tested to be equal to one. If NSUB equals one, the algorithm is exited and the value in NR equals the maximum distance from the current sonobuoy where a submarine detection can occur. If NSUB is not equal to one, it is set equal to one, NR is reset to a value of one less than it was for the last unsuccessful FOM, propagation loss test. Testing of FOM and propagation loss is then resumed with an index increment of one. If, during the original testing, NR becomes negative, it indicates there are no propagation loss values less than current buoy FOM. However, if the increment NSUB is greater than one, the last values of the propagation loss profile have been bypassed; so NSUB is set to one, NR is increased by LSUB and testing resumes. If NSUB goes negative again, it indicates no possible detection can be made by the current buoy on the present submarine run; so the program jumps to the point where DLARGE is set for the current buoy to negative one to indicate no possible detection.

E. TRSH SUBROUTINE (TASDA AC)

The two seed values, JJN and JJJ, were set to the original seed values, MYSEED and NBSEED, just prior to the call on PROB. This accomplishes seed reinitialization.

APPENDIX D

DICTIONARY OF TASDA WORDS IN COMMON

AC - Dimensioned as a logical word, used in AMAX subroutine as flag, if set to true, the subroutine TRSH will be called; if false, subroutine GEOM will be called.

AK1(6) - List of estimates of the probability of a single submarine detection for each given figure of merit.

AK2(6) - List of estimates of the probability of two simultaneous submarine detections. See AK1.

AK3(6) - List of estimates of the probability of three or more simultaneous submarine detections.

ANPTS - Floating point equivalent of NPTS. See NPTS. Set to NPTS in main program.

ANRUNS - Floating point equivalent of NRUNS. See NRUNS. Set to NRUNS in main program.

BAK1(6) - List of the optimum probability of detection values for single sonobuoy detection. Used in OPUT for printout variable. Contains PD1 value which is printed out for each geometry. One value for each FOM.

BAK2(6) - List of probabilities of two simultaneous detections associated with the geometry spacing which yields optimum single sonobuoy Pd. Used in OPUT as printout variable.

BAK3(6) - List of probabilities of three or more simultaneous detections. See BAK2.

BHTM1(6)- List which contains the mean holding time values of single sonobuoy detections. Set to HTM1 in OPUT for final printout values.

BHTM2(6)- List which contains the mean holding time values for two simultaneous sonobuoy detections. See BHTM1.

BHTM3(6)- List which contains the mean holding time values for three or more simultaneous detections. See BHTM1.

BSPF(6) - List of sonobuoy spacings assocaited with the calculated optimum Pd for each geometry. Set to SP x BUOYSP in OPUT. Used as spacing values which are printed for each geometry.

BTDM(6) - List which contains the values for mean time to first sonobuoy detection. Set in OPUT to TDM for final printout values for each geometry.

BUOYSP - Distance between first and second buoy of any geometry. Used to calculate minimum buoy spacing for given geometry.

C(16) - List of distances between submarine and each buoy in geometry $C(K) = (XB(K)-X_i)^2 + (YB_C(K)-Y_i)^2$. C changes each time submarine advanced 1 segment.

CD(6) - Coverage dimension used in area maximization to determine optimum size search area. Area will equal square with sides equal to 2 x CD.

CD1(6) - The best coverage dimension found for previously tested geometries. List is for different figures of merit.

CDD(6) - The coverage dimension currently being tested. See CD. List is for different figures of merit.

CDI(6) - Current increment of coverage dimension. List is for different figures of merit.

CDIO - Initial coverage dimension increment. Program controlled input parameter set to 25 in lab program.

CDMAX - Maximum coverage dimension; range 100 to 500 miles with 10 mile accuracy. Program controlled input parameter; set to 150 miles in lab program.

CDMIN - Minimum coverage dimension; range 5 to 50 miles with one mile accuracy. Program controlled input parameter set to 10 miles in lab program.

CDO - Initial coverage dimension; range from 25 to 150 miles. Program controlled input parameter set to 100 in lab program.

CDX - Approximate coverage dimension. Initially set in MAIN program, depending on the type of submarine input to program.

DF - Distance moved by target from starting coordinates before aircraft arrives on station.

DIRMAX - Maximum bearing angle from 0 to 360° in one degree increments. Used as operator controlled input parameter when target submarine is in holding type operations.

DIRMIN - Minimum bearing angle from 0 to 360° in one degree increments. See DIRMAX.

DISMAX - Maximum distance traveled by submarine from 0 to 500 nautical miles in one mile increments when moving in holding type operations. Operator controlled input parameter.

FM(6) - List containing figure of merit values. Operator controlled input parameter. High figure of merit corresponds to more easily detected submarine noise.

FM1 - Value of base figure of merit. Entered in program as user controlled input variable.

FMD(6) - Figure of merit differences; if greater than one figure of merit subset, DB difference is computed for each subset from base figure of merit. FM1 used in TRSH.

FMDB(16)- Figure of merit deviation for each buoy computed as a random normal deviation about a mean of zero and a standard deviation equal to SIGB.

FMDR - Figure of merit deviation for each run. Computed in same manner as FMDB.

FMF(16) - Final figure of merit calculated for each buoy in a pattern. Calculation takes place in PROB subroutine and equals FM + FMDB () + FMDR.

FMOT(6) - Figure of merit for printout when area maximization option used. Vector of six possible values set in AMAX subroutine.

GEOEND - Logical word, used in determining if all applicable geometries for submarine type and operation mode have been tested. If true, geometry subroutine is exited.

HOLD1(6)- Dimensioned as list of logical words. Used in PROB as flag to indicate at least one sonobuoy is holding contact with the submarine at current submarine position for FOM as indexed.

HOLD2(6)- See HOLD1. Flag indicates two or more sonobuoys are holding contact.

HOLD3(6)- See HOLD1. Flag indicates three or more sonobuoys are holding contact.

HT1(6,6)- Mean holding time for single sonobuoy detections. Used as part of search algorithm to store up to six different times corresponding to different buoy spacings for up to six different FOM.

HT2(6,6)- Mean holding time for two or more simultaneous sonobuoy detections. See HT1.

HT3(6,6)- Mean holding time for three or more simultaneous sonobuoy detections. See HT1.

HTM1(6) - List of sample mean holding times of single detections for each figure of merit (up to 6).

HTM2(6) - List of sample mean holding times of two simultaneous detections. See HTM1.

HTM3(6) - List of sample mean holding time of three simultaneous detections. See HTM1.

IB1 - Used as pointer or index to obtain the X and Y coordinates of the first sonobuoy in a geometry.

ICHAN - Set to 1/4 the number of monitoring channels available. Used to index monitoring cycles.

ICON - Dimensioned in data statement to (HC). Used in printout to indicate high confidence for mean time to detection.

IDENT(4,40) - Geometry identification code. One for each sonobuoy geometry. Alphanumeric charcters designating each geometry or pattern. Geometry data input parameter.

IDT(12) - Tactics identification. No longer used in TASDA MOD III.

IGE - Total number of geometries index. Set in INDAT subroutine to the number of geometries read into program. Used also as geometry index and set to geometry being tested or next geometry to be tested.

IGEA(6) - The index of the geometry yielding the best coverage used in area maximization subroutine.

IGFLG - Flag used in GEOM subroutine to indicate whether or not the geometry index has been chosen. If it equals two, index has been chosen; skip algorithm to find index. Flag set in TROP or AMAX prior to GEOM call.

II(6) - Used as counting variable in PROB. Incremented by one for each sonobuoy in contact with the submarine at current submarine position for indexed FOM. Initialized to zero after each submarine step.

INDALT(6) - Used to store time increment when submarine contact last held, plus a set time. When the current time increment exceeds the INDALT value, the FOM for each sonobuoy is decreased by two, indicating the operator is no longer alerted.

IOFLG - Flag used in OPUT to determine which portion of subroutine is to be exercised. IOFLG = 3 indicates the set up part of OPUT is to be exercised; IOFLG = 1 indicates printout portion is to be exercised. Set in TROP or AMAX prior to OPUT call.

IOUT - Flag used in OPUT to indicate which probability subroutine used, PROB or PROBF. IOUT = 1 indicates PROBF used; so calculate extra data.

ISEED(6) - Set to current seed value of random number generator prior to calls on PROBF. Seed restored to this value in OPUT, if PROBF has been used.

ISPEED - Option variable: 0 = Use slow method of computation of probability of submarine detection (subroutine PROB); 1 = more rapid method. Use (subroutine PROBF). Operator controlled input parameter.

ISTEP - Truncated integer value of STEP. Set in MAIN to STEP.

ITM - Set to current M index value in OPUT to indicate extra calculations completed for this M value. Tested in AMAX to determine if extra calculations needed in OPUT.

IZONE4(4,200) - Array containing four buoy subsets of a geometry. Each subset designates the monitoring sequence of buoys if a four-channel processor is in use. Subsets are identified by buoy position in a given geometry.

IZONE8(8,120) - Array containing eight buoy subsets of geometries. See IZONE4.

IZONEC(12,100) - Array containing twelve buoy subsets of a geometry. See IZONE4.

J - Common index value. Used throughout program.

JBUOY(41) - List which contains the positions of the X and Y coordinates of the first buoy of i^{th} geometry in the KXB and KYB lists, i.e., if JBUOY(3) = 15, then the X and Y coordinates of the first buoy of the third geometry are located in KXB(15) KYB(15).

JCOUNT - Used as counter to determine number of calls made on random number generator subroutine (RADN). Sensitivity analysis aid.

JFMI(6,6) - Figure of merit index used in search algorithm. Is used to keep track of which FOM the current Pd and buoy spacing are associated. Initialized in TROP and AMAX. Used in REFN.

JJ1(6) - List of flags used in PROB. Initialized to zero for each submarine run. Set to one to indicate single sonobuoy contact made with submarine at indexed FOM.

JJ2(6) - See JJ1. Set to one to indicate two simultaneous sonobuoy contacts made on submarine during current run.

JJ3(6) - See JJ1. Set to one to indicate three or more simultaneous sonobuoy contacts made on submarine during current run.

JJN - Current seed value for random number generator. Initialized to original seed value at start of program. Changes each time random number generator is used.

JJNO - The original seed value for pseudo-random number generator. NADC laboratory program uses 8063. Program controlled input parameter.

JPLOT(41) - List which contains the position of the first character of the format statement for the i^{th} geometry. Characters for format statements are in THE TABLE IPLOT. See JBUOY.

JPM(6,6) - Spacing index used in search algorithm. Contains values which associate buoy spacing to Pd and FOM. Initialized in TROP or AMAX. Used in REFN.

JZONE(41,3) - Array containing the positions of the first four, eight, or 12 buoy zone of the i^{th} geometry; i.e., if JZONE(3,2) = five, then the first eight buoy zone of the third geometry would start at IZONE8(5).

JZONEO - Index set to value which indicates the starting position of the current geometry and buoy zone. Used in GEOM to assist in setting up monitoring times of sonobuoys.

JZONE1 - Index set to value which indicates the last position of the current geometry and buoy zone. Used in GEOM to indicate the last buoy zone of the current geometry has been reached.

JZONET(12) - Set to values of monitoring times for monitoring zone of geometry in use for PRINTOUT. Used in OPUT.

K - Used as buoy coordinate index of current geometry in probability calculations and used in REFN to associate best previous Pd with its buoy spacing.

KGC(17,2,2) - Array for geometry classification. Contains position in KGI where first applicable geometry index is located KGC(A,B,C,) A = number of buoys in geometry, B = type of submarine, C = submarine motion: i.e., KGC(8,1,1) = 16 would indicate for the first applicable geometry having eight sonobuoys to be used against a conventional-holding submarine would be found in KGI(16).

KGI(99) - List containing indices of buoy geometries to be used for given inputs.

KK - Index variable. Used in REFN to designate probability of detection associated with latest buoy spacing tested. KK index value set in TRSH or AMAX.

KKK - Constant used as a multiplier in random number generator. Set to 99947 for IBM 360 computer random number generator. Program controlled input parameter.

KN - Equals the number of applicable buoy geometries in current classification set. When not equal to zero, I1 points to an applicable geometry index used to select appropriate geometries.

KON1 - Used as a constant equal to one. Set to one in program initialization process.

KON2 - See KON1. Set to two in data statement.

KON3 - See KON1. Set to three in data statement.

KXB(480) - Array which contains the X coordinate of all buoys of all geometries. Average of 12 buoys for 40 geometries. Index pointers are used to select first buoy of given geometry.

KYB(480) - Array which contains the Y coordinate for all buoys of all geometries. See KXB.

L - Index variable used for various indexing throughout program.

LAS(6) - Contains value indicating current state of search algorithm for area maximization problem. Vector used for different FOM values.

LCON - Dimensioned in data statement to alphanumeric; (LC). Used in printout to indicate low confidence.

LFA(6) - Flag indicating if coverage dimension is at limits. Flag set if CDD = CDMIN or CDD = CDMAX. List is for different figure of merit.

LIM(6) - Dimensioned to logical word. Used in search algorithm to indicate lower or upper boundary of buoy spacing has been reached. Vector for six different FOM.

LIM1 - Dimensioned to logical word. Used in search algorithm to indicate lower or upper bound of the spacing increment has been reached.

LPT - Option variable which determines problem type: 1 = threshold PROB; 2 = spacing optimization; 3 = area maximization. Operator controlled input parameter.

LS(6) - Indicates state of search algorithm. 0 = initial; 1 = one set of runs made at original spacing; 2 = all probabilities computed thus far monotonic increasing/decreasing; 3 = approximate to optimum reached P(SP-SPI) $\leq$P(SP+SPI); 4 = closer to optimum, i.e., checked

P(SP) = P(SP + 2SPI); 5 = same as two, but at minimum or maximum limit; 6 = closer to limit; 8 = P > threshold; 9 = P meets stop criteria.

LSA(6) - Flag indicating whether most recently computed statistics are to be saved. LSA = 0 implies most recent computed probabilities are greater than THRESH or greater than all earlier probabilities; so save statistics. List for different FOM.

LT - Indicator value which determines which test is conducted in REFN subroutine: LT = 1, threshold test; LT = 2, comparison test; LT = 3, optimum test. Value set in TRSH or AMAX prior to REFN call.

M - Index variable normally used to index the FOM currently being looked at, but used throughout program for many indices.

M1 - Index variable used in conjunction with M.

MAXB - Operator controlled input parameter. The maximum number of sonobuoys allowed for a buoy pattern. Used in determining which sonobuoy patterns will be tested. Range from 0 to 16 sonobuoys.

MCON - Dimensioned in data statement to (MC). Used in printout to indicate medium confidence.

MFM - Operator controlled input parameter. Sets the number of FOM inputs which program will use. Values from 1 to 6 are allowed.

MINB - Operator controlled input parameter. Indicates minimum number of sonobuoys allowed for a sonobuoy pattern. See MAXB range from 0 to 16 sonobuoys.

MN - Monitoring cycle indicator. Set in GEOM to indicate whether there is continuous sonobuoy monitoring or a monitoring schedule. MN = 2 indicates continuous monitoring; MN = 0 indicates monitoring schedule.

MODE - Code word for type of submarine being searched. 0 = conventional; 1 = nuclear. Program input parameter.

MON(16) - Flag for each buoy in a geometry to indicate if buoy is being monitored continuously or on a monitor schedule. Continuous monitoring occurs when number of channels is greater than number of buoys. MON set to 2 for continuous monitoring.

MYSEED - Random number generator seed. Equals 1333567.

NA - Used as flag indicator. Set in REFN to 1, if Pd is greater than threshold, if current Pd is greater than last Pd, or if optimum Pd has been found. Tested in TRSH and AMAX.

NB - Number of buoys in given sonobuoy pattern or geometry. Geometry data input parameter used as index in subroutine INDAT.

NCHAN - Operator controlled input parameter. Indicates the number of processing channels available in aircraft. Numbers 4, 8, 12, 16 are allowed. Used to determine monitoring cycle of buoys when buoys in pattern outnumber processing channels.

NDET - Dimensioned in data statement to (ND). Used in printout to indicate NO DETECTION.

NF(6) - Index relating FOM to FOM subset; i.e., if six FOM are inputs to program, the first three could be in subset 1; so NF(1) = 1, NF(2) = 1, and NF(3) = 1.

NFD(6) - List containing the number of runs in which at least one submarine detection occurred for each FOM. Initialized to zero for each new geometry.

NFM - Number of FOM. See MFM.

NFMS - Number of FOM subsets. Used as index when calculating optimum spacing ratios for different FOM subsets.

NHOWMV - Input parameter denoting type of submarine movement; 1 = holding (on station); 2 = transiting; 3 = frontal coverage.

NHT1(6) - List of the number of single sonobuoy detections for each FOM for a given geometry. Initialized to zero for each new geometry.

NHT2(6) - List of the number of two simultaneous detections on submarine for a given geometry. See NHT1.

NHT3(6) - List of the number of three simultaneous detections on a submarine. See NHT1.

NNFD(6) - Printout variable used in OPUT set to LCON, MCON, or ICON (low, medium, or high confidence), as tag to mean time to first detection printout.

NOW(400,16) - Flag for each buoy for each segment of a run to determine whether buoy is being monitored. 0 = no-monitoring, 1 = monitoring.

NPLIND - Code word which denotes how propagation loss profile is obtained for program. 0 = formula, 1 = use of actual propagation loss data. Operator controlled input parameter.

NPMAX - Maximum number of characters in geometry plot statement. Set 160 in INDAT subroutine (not used unless desire printout of what geometries look like).

NPTS - Operator controlled input parameter designates the number of values input from propagation loss program from 1 to 205.

NRUNS - Input variable determines the number of submarine runs for each probability calculation. Program controlled input parameter. Set to 10 for NADC laboratory program.

NSEGMX - Maximum number of segments a submarine can move for given input parameters; if greater than NSGMAX, see NSGMAX, then program aborts and abort alert is displayed.

NSGMAX - Maximum number of segments submarine is allowed to move in any run. Used as comparison for abort criteria. Abort if calculated number of segments greater than NSGMAX. Program controlled input parameter set to 400.

P1(6,6) - Used in search algorithm to contain Pd values for up to six different sonobuoy spacings for six different FOM indexing scheme of search algorithm. Keeps track of greatest value for choosing best spacings. Used in TRSH, REFN, and OPUT.

P2(6,6) - See P1. Contains Pd of two simultaneous detections associated with indices of P1.

P3(6,6) - See P1. Contains Pd of three or more simultaneous detections associated with indices of P1.

PI - Mathematical constant. PI = 3.141592654. Program controlled input parameter.

PL(16) - List of propagation losses corresponding to sonobuoy submarine detection for each sonobuoy for current submarine position. If NPLIND = 1, it indicates formula used: $PL(K) = 17\ LOG_{10}(C(K)) + 66$.

PLC(205) - Array containing values of propagation loss. Computed by propagation loss program. Used to compute the maximum distance from a submarine in which a sonobuoy can detect the submarine.

PSDIF - Probability stop difference. Program controlled input parameter used to stop search for solution on given sonobuoy pattern. Stops search when estimated maximum probability and the largest value found are less than PSDIF. Range 0 to .2; set to .08 in laboratory program.

RADIUS(40)- Sonobuoy geometry radius equals distance from geometry origin to most distant sonobuoy in pattern. Geometry data input parameter.

RFRNGE - Radio frequency range of sonobuoys from 50 to 250 miles. Used to compute maximum spacing between sonobuoys in different patterns. Operator controlled input.

SIGB - Standard deviation of normal distribution for minimum detectable submarine noise by a sonobuoy. Program controlled input parameter.

SIGR - Standard deviation of normal distribution for minimum detectable submarine noise for each submarine run. (Run Increment)

SP(6) - Spacing ratio list of spacing ratios for each FOM. Used to convert X and Y buoy coordinates in grid units to coordinates in nautical miles. Threshold subroutine finds SP for which detection probability exceeds an input value.

SP1(6) - The spacing ratio corresponding to the best coverage dimension found thus far in program that yields threshold probabilities greater than THRESH.

SPF(6) - Final buoy spacing for optimum single buoy Pd. Not used in TASDA MOD3, but has not been removed from COMMON.

SPGRAN - Spacing granularity measure in miles; range 0 - 20 miles with accuracy of one mile. Used in determining the minimum spacing ratio. Program controlled input parameter set to 5 in laboratory program.

SPGRH(40) - Spacing grid parameter holding. Used with CD, coverage dimension, to calculate geometry spacing ratio which, in turn, is used with sonobuoy grid coordinates to determine sonobuoy X and Y coordinates. Geometry data input parameter.

SPGRT(40) - Spacing grid parameter transiting. See SPGRH.

SPI(6) - Spacing ratio increment. The amount the spacing ratio is changed for the next spacing ratio to be tested. List corresponds to different FOM.

SPIGRH(40) - Spacing increment grid parameter holding used with CD, coverage dimension, to calculate initial geometry spacing grid increment which, in turn, is used to calculate new sonobuoy X and Y coordinates. Geometry data input parameter.

SPIGRT(40) - Spacing increment grid parameter transiting. See SPIGRH.

SPMAX - Maximum allowable buoy spacing of a geometry. Obtained by dividing RFRNGE by geometry RADIUS.

SPMIN - Minimum allowable buoy spacing of a geometry.

SPO(6) - The spacing ratio corresponding to the best coverage dimension found thus far in program that yields optimal probabilities which are less than THRESH.

STEP - Set to STEPB or STEPL, depending on submarine speed and maximum distance it can move. Time increment used to calculate successive submarine positions.

STEPB - Large time increment. Used to calculate successive submarine positions along track. Range is three to five minutes with one minute accuracy. Program controlled input parameter. Set to four minutes in laboratory program.

STEPL - Small time increment. See STEPB. Range one to three minutes with .1 minute accuracy. Set to two minutes in laboratory program.

T1 - Lower printout probability threshold. Program controlled input parameter; set to .20 in laboratory program.

T2 - Upper printout probability threshold: See T1. Program controlled input parameter; set to .75 in laboratory program.

TD(6,6) - Storage matrix for mean time to detection values associated with different buoy spacings and FOM. Used in search algorithm to keep track of these values.

TDM(6) - List of sample mean times to first detection for each given figure of merit (up to six). Equals zero, if no detections occur.

THRC - Threshold probability default. If THRESH not specified by operator, THRESH = THRC. Program controlled input parameter. Set to .99 in laboratory program.

THRESH - Threshold probability. Operator controlled input parameter used to determine probability of detection for sonobuoy geometries. If threshold probability is exceeded, criteria is met to stop search algorithm.

TMN(40,3)- Time to monitor buoys. Contains the amount of time any zone of buoys are to be monitored; i.e., TMN(6,2) = 10.0 would indicate the eight buoy zones for the sixth geometry should each be monitored for ten minutes after which the next eight buoy zone should be monitored. Set by geometry input data.

TMON - Monitoring time used in calculations, taken from TMN array. See TMN.

TSNMAX - Operator controlled input parameter. Maximum time target submarine can snorkel measured in minutes from 30 to 120 minutes. Used in determining snorkel and submerged cycle of submarine.

TSNMIN - Operator controlled input parameter minimum time target submarine can snorkel measured in minutes from 15 to 60 minutes. See TSNMAX.

TSUMAX - Operator controlled input parameter. Maximum submerged time of target submarine measured in minutes from 360 to 720 in ten minute increments. See TSNMAX.

TSUMIN - Operator controlled input parameter. Minimum submerged time of submarine measured in ten minute increments from 120 to 240. See TSNMAX.

VELM - Submarine submerged speed in nautical miles/minute. Obtained by dividing input parameter VEL by 60. Range: 1/15 to 2/3 knots per minute.

VELSNM - Submarine speed while snorkeling in nautical miles/min. Obtained by dividing input parameter VELSNK by 60. Range: 1/20 to 1/5 knot per minute.

XB(16) - Array of X coordinates of sonobuoys in current geometry. One to 16 buoys available in any given geometry.

XF1 - First X coordinate of rectangle in which submarine completes movement. Used with XF2, YF1, YF2 to specify area chosen at random. Operator controlled input parameter. Range: -1000 to +1000 nautical miles.

XF2 - Second X coordinate of rectangle in which submarine completes movement. See XF.

XS1 - First X coordinate of rectangle in which submarine movement originates. Used with XS2, YS1, YS2 to specify area in which submarine movement originates. Actual point within area is chosen at random. Operator controlled input parameter. Range: -1000 to +1000 nautical miles.

XS2 - Second X coordinate of rectangle in which submarine movement originates. See XS1.

YB(16) - Array of Y coordinates of sonobuoys in current geometry. See XB.

YF1, YF2- Y coordinates of rectangle in which submarine movement terminates. See XF1.

YS1, YS2- Y coordinates of rectangle in which submarine movement originates. See XS1.

COMPUTER PROGRAM LISTING A

TASDA MOD III PROGRAM

```
      BLOCK DATA
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TOM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SF1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),        LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JEUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
```

```
     *         DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,      044
     *         TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,        045
     *         IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,EUCYSP,JZCNEC,JZCNE1         046
      COMMON/A/ M,ISTEP,      J,K                                            047
      COMMON/A/ IOFLG,IGFLG                                                  048
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA                     049
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1             050
      COMMON/A/ LFA(6)                                                       051
      COMMON /A/ESPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),     052
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMOT(6),      053
     2          NNFD(6), JJN                                                 054
      COMMON/B/ JCOUNT, KKK, MYSEED
C                                                                            055
C        MAXIMUM NO. OF SEGMENTS PER RUN IS 400                              056
      DATA NSGMAX /400/                                                      057
C        TASCA  BUILT-IN CONSTANTS                                           058
C        SIGR,SIGB ARE TEMPORAL AND SPATIAL SIGMAS FOR RANDOMIZING FM        059
C        PSDIF = 8*(SMALLEST VALUE OF DIFFERENCE BETWEEN ESTIMATED           060
C                  MAXIMUM AND CURRENT VALUE WITHOUT QUITTING)               061
      DATA SIGR /5./, SIGB /3./, PI /3.141592654/ ,THRC /.99/,               062
     *     STEPL /2./, STEPB /4./, KCN1 /1/, KCN2 /2/, KCN3 /3/              063
      DATA SPGRAN /5./, PSDIF /.08/, CDMIN /10./, CDO /100./,                064
     1     CDIO /25./, CDMAX /150./, NRUNS / 10/, JJNO /8063/                065
      DATA NDET /4H(ND)/, LCON /4H(LC)/,MCON /4H(MC)/, ICON /4H(HC)/         066
      DATA T1 /.20/, T2 /.75/                                                067
      DATA P1 /36*-1.0/, P2/36*-1.0/,P3 /36*-1.0/,
     *HT1 /36*-1.0/, HT2 /36*-1.0/, HT3 /36*-1.0/, TD /36*-1.0/,
     *CDD /6*1.0/
      DATA MYSEED /1333567/, KKK /99947/, JCOUNT /0/
      END                                                                    077
CASW COMPUTER SIMULATION PROGRAM FOR TASCA                                   080
C                                                                            081
C                                                                            082
C        MAX. NO. OF BUOYS IS 16                                             083
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),     084
     1          SE(6,16),MCN(16)                                             085
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6                                086
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),      087
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),        088
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),          089
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),          090
     *          NHT3(6),NF(6),         LSA(6),CD1(6)                         091
C        NUMBER OF POSITION INDEXES IS 6                                     092
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),          093
     *          HT3(6,6),TD(6,6)                                             094
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205              095
      COMMON/A/ PLC(205)                                                     096
      COMMON/A/ JZONET(12)                                                   097
```

```
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C         AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C         AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C         AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KCN1,KCN2,KCN3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,BUOYSP,JZONEC,JZONE1
      COMMON/A/ M,ISTEP,        J,K
      COMMON/A/ ICFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2          NNFD(6), JJN
      COMMON/B/ JCOUNT, KKK, MYSEED
C
C
C
      JJN = MYSEED
      RN = 0.0
      CALL INDAT
  998 FORMAT (8F10.3)
  999 FORMAT(9I5)
      READ (5,999) LPT,MODE,NHOWMV,NFM,NPLIND,NPTS,MINB,MAXB,NCHAN
      MFM = NFM
      READ (5,998) TMELTE,RFRNGE
      READ (5,998) VEL,VELSNK,TSNMIN,TSNMAX,TSUMIN,TSUMAX
```

```
      READ (5,998) XS1,YS1,XS2,YS2,XF1,YF1,XF2,YF2
      READ (5,998) HDEPTH,SDEPTH,(FM(I),I=1,NFM)
      IF (NPLINC .NE. 1) GO TO 10
      IF (NPTS .LE. 205) GO TO 9
      WRITE (6,2032) NPTS
2032  FORMAT (3X, I5)
      GO TO 66
9     READ (5,998) (PLC(I),I=1,NPTS)
10    IF (LPT .EQ. 2) GO TO 20
      READ (5,998) THRESH
      GO TO 30
20    THRESH = THRC
   30 CONTINUE
      READ (5,2033) ISPEED
2033  FORMAT (I5)
      WRITE (6,997) LPT, MODE, NHCWMV, NFM, NPLINC, NPTS, MINB , MAXB,
     1               NCHAN, TMELTE, RFRNGE, VEL, VELSNK, TSNMIN, TSNMAX,
     2               TSUMIN, TSUMAX, XS1, YS1, XS2, YS2, XF1, YF1, XF2,
     3               YF2, THRESH, ISPEED, (FM(I),I = 1,NFM)
997   FORMAT (1X,5HLPT =,I5,3X,6HMODE =,I5,3X,8HNHCWMV =,I5,3X,5HNFM =,
     1        I5,3X,8HNPLIND =,I5,3X,6HNPTS =,I5,3X, 6HMINE =,I5,3X,
     2        6HMAXB =,I5,/,1X,7HNCHAN =,I5,3X,8HTMELTE =,F10.2,3X,
     3        8HRFRNGE =,F10.2,/,1X,5HVEL =,F10.2,2X,8HVELSNK =,F10.2,
     4          2X,8HTSNMAX =,F10.2,2X,8HTSNMAX =,F10.2,2X,8HTSUMIN =,
     5        F10.2,2X,8HTSUMAX =,F10.2,2X,/,2X,20HSTARTING COORDINATES,
     6        /,4F20.2,/,1X,18HENDING COORDINATES,/,4F20.2/,1X,
     7        8HTHRESH =,F10.2,3X,8HISPEED =,I5,/,1X,15HFIGURE OF MERIT,
     8        /,6F20.2)
C
C        END OF DATA INITIALIZATION
C
      ANPTS=NPTS
      ANRUNS=NRUNS
      VELM=VEL/60.0
      VELAVG = VEL
      IF(MODE.EQ.1)GO TO 84
      VELSNM=VELSNK/60.0
      TSUAVG = .5*(TSUMAX + TSUMIN)
      TSNAVG = .5*(TSNMAX + TSNMIN)
      CYCAVG=TSUAVG+TSNAVG
      VELAVG = (VEL*TSUAVG + VELSNK*TSNAVG)/CYCAVG
   84 CONTINUE
C        DF = APPROXIMATE DISTANCE MOVED BY TARGET BEFORE AIRCRAFT
C        ARRIVES ON STATION
      DF = TMELTE*VELAVG*.0166666667
      IF (LPT .EQ. 3) GO TO 101
      GO TO (85, 90, 95), NHCWMV
85    DIRMAX = XF1
```

```
      DIRMIN = YF1
      DISMAX = XF2
      CDX = .5*AMAX1(ABS(XS1 - XS2), ABS(YS1 - YS2)) + DF + DISMAX
      GO TO 100
90    CDX1 = AMAX1(XS1,XS2,XF1,XF2) - AMIN1(XS1,XS2,XF1,XF2)
      CDX2 = AMAX1(YS1,YS2,YF1,YF2) - AMIN1(YS1,YS2,YF1,YF2)
      DISMAX = SQRT(CDX1*CDX1 + CDX2*CDX2) - DF
      CDX = .5*AMAX1(CDX1,CDX2)
      GO TO 100
95    CDX = .5*XS1
      DISMAX = YS1
      XS1 = -CDX
      XS2 = CDX
      YS1 = XS2 + DF
      YS2 = YS1
      XF1 = XS1
      XF2 = XS2
      YF1 = XF1
      YF2 = YF1
      GO TO 100
101   DISMAX = XS1
C
C RUNT=AVERAGE TRANSIT TIME WHEN TARGET MOVES DISMAX
100   RUNT = DISMAX/VELAVG
      STEP = STEPL
      IF(RUNT.GT.5.0) STEP = STEPB
      ISTEP = STEP
      IF(MODE.EQ.0)GO TO 36
      NSEGMX = (RUNT*60.)/STEP + 10.
      GO TO 27
36    IF(VELSNM .GT. VELM)GO TO 361
      TSUM = TSUMIN
C FOR MAXIMUM NO. OF SEGMENTS MINIMIZE SUBMERGE TIME AND MAXIMIZE
C SNORKEL TIME
      TSNM = TSNMAX
      GO TO 362
361   TSUM = TSUMAX
C FOR MAXIMUM NO. OF SEGMENTS MINIMIZE SNORKEL TIME AND MAXIMIZE
C SUBMERGE TIME
      TSNM = TSNMIN
362   NTSUM = TSUM/STEP
      TSUM = NTSUM*ISTEP
      NTSNM = TSNM/STEP
      TSNM = NTSNM*ISTEP
      CYCDUM = TSUM + TSNM
      NSEGMX = DISMAX*CYCDUM/(STEP*(VELM*TSUM + VELSNM*TSNM)) + 10.
C
27    IF (NSEGMX .LE. NSGMAX) GO TO 271
```

```
C       ABORT ALERT
        GO TO 66
271     CONTINUE
        ICHAN = NCHAN/4
        IF (ICHAN.EQ.4) ICHAN=3
170     KN = 0
        IOFLG = 0
        WRITE (6,1000) NSEGMX, KN, IOFLG, LPT
1000    FORMAT (5X,8HNSEGMX =,I5,4HKN =,I5,7HIOFLG =, I5,5HLPT =, I5)
        GO TO (180, 180, 9000), LPT
180     CALL TROP
        GO TO 66
9000    CALL AMAX
66      WRITE (6,9997)
9997    FORMAT (11H END OF RUN)
        WRITE(6,8866) JCOUNT
8866    FORMAT('  THE NUMBER OF TIMES THE RANDOM NUM. GEN. CALLED = ',I9)
        STOP
        END
        SUBROUTINE TROP
C
C          MAX. NO. OF BUOYS IS 16
        COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1            SE(6,16),MCN(16)
C          MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
        COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *            INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *            SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *            CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *            NHT3(6),NF(6),       LSA(6),CD1(6)
C          NUMBER OF POSITION INDEXES IS 6
        COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *            HT3(6,6),TD(6,6)
C          MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
        COMMON/A/ PLC(205)
        COMMON/A/ JZONET(12)
C          NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
        COMMON/A/ IDT(12)
        COMMON/A/ KGC(17,2,2)
C          MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C          (INCLUDING REPETITIONS) IS 99
        COMMON/A/ KGI(99)
C          MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
        COMMON/A/ IDENT(4,40)
C          MAXIMUM NO. OF GEOMETRIES IS 40
        COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SFIGRT(40),SPGRH(40),
     1            SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C          AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
```

```
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNC,NDET,LCCN,MCCN,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MCDE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NFCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,EUCYSP,JZONEO,JZONE1
      COMMON/A/ M,ISTEP,     J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMOT(6),
     2          NNFD(6), JJN
C
C
C        INITIALIZATION FOR THRESHOLD/OPTIMUM PROBLEM
C
184   IGFLG =1
      CALL GEOM
185   IF (GEOEND) RETURN
      IF (NHOWMV .GE. 2) GO TO 187
      SF(1) = SPGRH(IGE)*CDX
      SFI(1) = SPIGRH(IGE)*CDX
      GO TO 188
187   SF(1) = SPGRT(IGE)*CDX
      SFI(1) = SPIGRT(IGE)*CDX
188   DO 210 K=1,4
210   JFM(1,K) = K
      DO 220 K=1,MFM
220   JFMI(1,K) = K
      NF(1) = MFM
      NFMS = 1
      CALL TRSH
      IF (IOFLG .LT. 0) RETURN
C
13    DO 250 M=1,NFMS
      K = 2
      IF (LS(M) .EQ. 8) K = 4
```

```
      K = JPM(M,K)
      NFM = NF(M)
      ICUT = 0
      IF (ISPEED .EQ. 1) IOUT = 1
      DO 244 N=1,NFM
      J = JFMI(M,N)
      L = J
      ICFLG = 3
      CALL OPLT
244   CONTINUE
250   CONTINUE
C
      ICFLG = 1
      CALL OPLT
310   GO TO 184
      END
      SUBROUTINE PROB
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TOM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),        LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SFGRT(40),SFIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
```

```
C       AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
        COMMON/A/ IZONE8(8,120)
C       AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
        COMMON/A/ IZONEC(12,100)
        COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
      *           KON1,KON2,KON3,SPGRAN,FSDIF,CDMIN,CDMAX,CDC,CDIO,
      *           NRUNS,JJNC,NDET,LCCN,MCCN,ICON,T1,T2
        COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
      *           MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
      *           DIRMIN,DIRMAX,DISMAX,NFOWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
      *           TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
      *           IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,EUCYSP,JZONEC,JZONE1
        COMMON/A/ M,ISTEP,        J,K
        COMMON/A/ ICFLG,IGFLG
        LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
        COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
        COMMON/A/ LFA(6)
        COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
      1           BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
      2           NNFD(6), JJN
C
C
C
81      DO 150 KXY=1,NFM
        NFD(KXY) = 0
        TDM(KXY) = 0.
        NHT1(KXY) = 0
        NHT2(KXY) = 0
        NHT3(KXY) = 0
        HTM1(KXY) = 0.
        HTM2(KXY) = 0.
        HTM3(KXY) = 0.
        AK1(KXY)=0.0
        AK2(KXY)=0.0
150     AK3(KXY)=0.0
        DO 140 I = 1,NRUNS
        DO 160 LXY=1,NFM
        HOLD1(LXY) = .FALSE.
        HOLD2(LXY) = .FALSE.
        HOLD3(LXY) = .FALSE.
        JJ1(LXY)=0
        JJ2(LXY)=0
160     JJ3(LXY)=0
        CALL RADN (RN, JJN)
        XS=XS1+RN*(XS2-XS1)
        CALL RADN (RN, JJN)
        YS=YS1+RN*(YS2-YS1)
        IF(NFOWMV.EQ.1)GO TO 16
```

```
      CALL RADN (RN, JJN)
      XF=XF1+RN*(XF2-XF1)
      CALL RADN (RN, JJN)
      YF=YF1+RN*(YF2-YF1)
CCALCULATE TARGET BEARING WITH X AXIS-THPX(THROUGH STEP 18)
      IF((XF.GT.XS).AND.(YF.EQ.YS))GO TO 11
      IF((XF.LT.XS).AND.(YF.EQ.YS))GO TO 12
      IF((XF.EQ.XS).AND.(YF.GE.YS))GO TO 1
      THP=ATAN (ABS((XF-XS)/(YF-YS)))
      IF((XF.GT.XS).AND.(YF.GT.YS))GO TO 2
      IF((XF.GE.XS).AND.(YF.LT.YS))GO TO 3
      IF((XF.LT.XS).AND.(YF.LT.YS))GO TO 4
      IF((XF.LT.XS).AND.(YF.GT.YS))GO TO 5
    1 TH=0.0
      GO TO 7
    2 TH=+THP
      GO TO 7
    3 TH=PI-THP
      GO TO 7
    4 TH=PI+THP
      GO TO 7
    5 TH=2.0*PI-THP
      GO TO 7
   11 TH=PI/2.0
      GO TO 7
   12 TH=3.0*PI/2.0
      GO TO 7
    7 IF(TH.LE.6.29)GO TO 8
      ICFLG = -1
      RETURN
    8 THY=TH*57.2957795
      THPX=(PI/2.0)-TH
      GO TO 28
16    CALL RADN (RN, JJN)
      THY=DIRMIN+RN*(DIRMAX-DIRMIN)
      THPX=PI/2.0-THY/57.2957795
   28 CONTINUE
      DIS=DISMAX
      BC=COS(THPX)
      BS=SIN(THPX)
      XS = XS + BC*DF
      YS = YS + BS*DF
      IF((NHOWMV .NE. 2).OR. LPT .EQ. 3) GO TO 18
      DIS = SQRT((XF-XS)*(XF-XS) + (YF-YS)*(YF-YS))
   18 CONTINUE
      NSEG=(DIS/VELM)/STEP+1.0
      IF(MODE.EQ.1)GO TO 71
      CALL RADN (RN, JJN)
```

```
      TSN=TSNMIN+RN*(TSNMAX-TSNMIN)
      NTSN=TSN/STEP
      TSN=NTSN*ISTEP
      CALL RADN (RN, JJN)
      TSU=TSUMIN+RN*(TSUMAX-TSUMIN)
      NTSU=TSU/STEP
      TSU=NTSU*ISTEP
      CYC=TSU+TSN
      CALL RADN (RN, JJN)
      BEGCYC=RN*CYC
      NSEG=(DIS/(VELM*STEP*TSU/CYC+VELSNM*STEP*TSN/CYC))+1.0
CDELTAX,DELTAY=TARGET MOVEMENT PER TIME INCREMENT
71    DELTAX=BC*VELM*STEP
      DELTAY=BS*VELM*STEP
      IF(MODE.EQ.1)GO TO 94
      DELXSN=BC*VELSNM*STEP
      DELYSN=BS*VELSNM*STEP
   94 CONTINUE
      SUM=0.0
      DO 96 J=1,12
      CALL RADN (RN, JJN)
96    SUM = SUM + RN
      FMDR=SIGR*(SUM-6.0)
      DO 99 K=1,NB
      SUM=0.0
      DO 40 L=1,12
      CALL RADN (RN, JJN)
   40 SUM=SUM+RN
      FMDB(K)=SIGB*(SUM-6.0)
CFMF(K)=FIRST INPUT FM(ADJUSTED FOR RANDOMNESS)
99    FMF(K)=  FM1+FMDR+FMDB(K)
98    DO 190 IXZ=1,NFM
  190 INDALT(IXZ)=0
      DO 50 J=1,NSEG
      DO 70  NXY=1,NFM
70    II(NXY)=0
      AJ = J
      TREAL=AJ*STEP
      IF(MODE.EQ.1)GO TO 31
      IA=TREAL/CYC
      AIA=IA
      TIC=TREAL-AIA*CYC+BEGCYC
      IF(TIC.GT.CYC)TIC=TIC-CYC
CTIC=ELASPED TIME OF PRESENT SUBMERGED/SNORKEL CYCLE
      IF(TIC.GE.TSU)GO TO 32
CMOVE TARGET(UPDATE XS,YS)
      XS=XS+DELTAX
      YS=YS+DELTAY
```

```
      GO TO 121
   32 XS=XS+DELXSN
      YS=YS+DELYSN
      GO TO 61
   31 XS=XS+DELTAX
      YS=YS+DELTAY
61    DO 60 N=1,NB
      IF(MON(N).EQ.2)GO TO 77
      IF(NOW(J,N).EQ.0)GO TO 60
77    C(N) = SQRT((XB(N)-XS)*(XB(N)-XS) + (YB(N)-YS)*(YB(N)-YS))
      IF(NPLIND.EQ.0)GO TO 43
      IF(C(N).GE.ANPTS-1.0)GO TO 60
      IC=C(N)
      AIC=IC
      PL(N)=PLC(IC+1)+(PLC(IC+2)-PLC(IC+1))*(C(N)-AIC)
      GO TO 46
43    IF (C(N) .GT. 1.) GO TO 44
      PL(N) = 66.
      GO TO 46
44    PL(N) = 17.*ALOG10(C(N)) + 66.
CCOMPUTE SIGNAL EXCESS FOR EACH BUOY(GREATER THAN O=DETECT)
   46 SE(1,N)=FMF(N)-PL(N)
      DO 120 NX=1,NFM
      SE(NX,N)=SE(1,N)+FMD(NX)
CINCREASE SE BY 2 DB(OPERATOR ALERTED BY PREVIOUS DETECTION)
      IF(INDALT(NX).GE.J)SE(NX,N)=SE(NX,N)+2.0
  120 IF(SE(NX,N).GT.0.0)II(NX)=II(NX)+1
   60 CONTINUE
C
121   DO 134 IXY=1,NFM
      IA = II(IXY) + 1
      IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 240
      GO TO (128, 124, 122), IA
C         II .GE. 3
240   HOLD3(IXY)=.TRUE.
      HTM3(IXY) = HTM3(IXY) + STEP
      JJ3(IXY) = 1
C         II .GE. 2
122   HOLD2(IXY) = .TRUE.
      HTM2(IXY) = HTM2(IXY) + STEP
      JJ2(IXY) = 1
C         II .GE. 1
124   HOLD1(IXY) = .TRUE.
      HTM1(IXY) = HTM1(IXY) + STEP
      INDALT(IXY) = J + IFIX(20./STEP)
      IF (JJ1(IXY) .EQ. 1) GO TO 126
      TDM(IXY) = TDM(IXY) + TREAL
      NFD(IXY) = NFD(IXY) + 1
```

```
      JJ1(IXY) = 1
126   IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 134
      GO TO (130,130,132), IA
128   IF (.NOT. HOLD1(IXY)) GO TO 130
      NHT1(IXY) = NHT1(IXY) + 1
      HOLD1(IXY) = .FALSE.
130   IF (.NOT. HOLD2(IXY)) GO TO 132
      NHT2(IXY) = NHT2(IXY) + 1
      HOLD2(IXY) = .FALSE.
132   IF (.NOT. HOLD3(IXY)) GO TO 134
      NHT3(IXY) = NHT3(IXY) + 1
      HOLD3(IXY) = .FALSE.
134   CONTINUE
C
   50 CONTINUE
      DO 137 JXY=1,NFM
      IF (.NOT. HOLD1(JXY)) GO TO 51
      NHT1(JXY) = NHT1(JXY) + 1
      IF (.NOT. HOLD2(JXY)) GO TO 51
      NHT2(JXY) = NHT2(JXY) + 1
      IF (.NOT. HOLD3(JXY)) GO TO 51
      NHT3(JXY) = NHT3(JXY) + 1
51    CONTINUE
      IF(JJ3(JXY).EQ.1)GO TO 55
      IF(JJ2(JXY).EQ.1)GO TO 65
      IF(JJ1(JXY).EQ.1)GO TO 75
      GO TO 137
   55 AK3(JXY)=AK3(JXY)+1.0/ANRUNS
   65 AK2(JXY)=AK2(JXY)+1.0/ANRUNS
   75 AK1(JXY)=AK1(JXY)+1.0/ANRUNS
137   CONTINUE
  140 CONTINUE
      WRITE(6,666)HTM1(1),HTM2(1),HTM3(1)
 666  FORMAT(' HTM1 = ',F20.8,'HTM2 = ',F20.8,'HTM3 = ',F20.8)
      DO 155 IXY=1,NFM
      IF (NFD(IXY) .EQ. 0) GO TO 145
      TDM(IXY) = TDM(IXY)/FLOAT(NFD(IXY))
145   IF (NHT1(IXY) .EQ. 0) GO TO 155
      HTM1(IXY) = HTM1(IXY)/FLOAT(NHT1(IXY))
      IF (NHT2(IXY) .EQ. 0) GO TO 155
      HTM2(IXY) = HTM2(IXY)/FLOAT(NHT2(IXY))
      IF (NHT3(IXY) .EQ. 0) GO TO 155
      HTM3(IXY) = HTM3(IXY)/FLOAT(NHT3(IXY))
155   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PROBF
C
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MCN(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),         LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JFLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNC,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,BUOYSP,JZONEO,JZONE1
      COMMON/A/ M,ISTEP,       J,K
```

```
      COMMON/A/  IOFLG,IGFLG
      LOGICAL    GECEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/  GECEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/  LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1           BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2           NNFD(6), JJN
C
C
      DIMENSION NMDET(6)
81    DO 150 KXY=1,NFM
      NFD(KXY) = 0
150   AK1(KXY)=0.0
      DO 140 I = 1,NRUNS
      CALL RADN (RN, JJN)
      XS=XS1+RN*(XS2-XS1)
      CALL RADN (RN, JJN)
      YS=YS1+RN*(YS2-YS1)
      IF(NHOWMV.EQ.1)GO TO 16
      CALL RADN (RN, JJN)
      XF=XF1+RN*(XF2-XF1)
      CALL RADN (RN, JJN)
      YF=YF1+RN*(YF2-YF1)
CCALCULATE TARGET BEARING WITH X AXIS-THPX(THROUGH STEP 18)
      IF((XF.GT.XS).AND.(YF.EQ.YS))GO TO 11
      IF((XF.LT.XS).AND.(YF.EQ.YS))GO TO 12
      IF((XF.EQ.XS).AND.(YF.GE.YS))GO TO 1
      THP=ATAN (ABS((XF-XS)/(YF-YS)))
      IF((XF.GT.XS).AND.(YF.GT.YS))GO TO 2
      IF((XF.GE.XS).AND.(YF.LT.YS))GO TO 3
      IF((XF.LT.XS).AND.(YF.LT.YS))GO TO 4
      IF((XF.LT.XS).AND.(YF.GT.YS))GO TO 5
    1 TH=0.0
      GO TO 7
    2 TH=+THP
      GO TO 7
    3 TH=PI-THP
      GO TO 7
    4 TH=PI+THP
      GO TO 7
    5 TH=2.0*PI-THP
      GO TO 7
   11 TH=PI/2.0
      GO TO 7
   12 TH=3.0*PI/2.0
    7 IF(TH.LE.6.29)GO TO 28
      ICFLG = -1
      RETURN
```

```
16    CALL RADN (RN, JJN)
      TH = (DIRMIN + RN *(DIRMAX - DIRMIN))/57.2957795
28    THPX=(PI/2.0)-TH
      DIS=DISMAX
      BC=COS(THPX)
      BS=SIN(THPX)
      XS = XS + BC*DF
      YS = YS + BS*DF
      IF((NHOWMV .NE. 2).OR.(LPT .EQ. 3))GO TO 18
      DIS=SQRT((XF-XS)**2+(YF-YS)**2)
   18 CONTINUE
      NSEG=(DIS/VELM)/STEP+1.0
      IF(MODE .EQ. 1  ) GO TO 71
      CALL RADN (RN, JJN)
      TSN=TSNMIN+RN*(TSNMAX-TSNMIN)
      NTSN=TSN/STEP
      TSN=NTSN*ISTEP
      CALL RADN (RN, JJN)
      TSU=TSUMIN+RN*(TSUMAX-TSUMIN)
      NTSU=TSU/STEP
      TSU=NTSU*ISTEP
      CYC=TSU+TSN
      CALL RADN (RN, JJN)
      BEGCYC=RN*CYC
      NSEG=(DIS/(VELM*STEP*TSU/CYC+VELSNM*STEP*TSN/CYC))+1.0
      DELXSN=BC*VELSNM*STEP
      DELYSN=BS*VELSNM*STEP
CDELTAX,DELTAY=TARGET MOVEMENT PER TIME INCREMENT
71    DELTAX=BC*VELM*STEP
      DELTAY=BS*VELM*STEP
      SUM=0.0
      DO 97 J=1,12
      CALL RADN (RN, JJN)
97    SUM = SUM + RN
      FMDR=SIGR*(SUM-6.0)
      DO 99 K=1,NB
      SUM=0.0
      DO 40 L=1,12
      CALL RADN (RN, JJN)
   40 SUM=SUM+RN
      FMDB(K)=SIGB*(SUM-6.0)
CFMF(K)=FIRST INPUT FM(ADJUSTED FOR RANDOMNESS)
99    FMF(K)=  FM1+FMDR+FMDB(K)
      DO 45  L = 1,NFM
45    NMDET(L) = 0
      IF (MODE .EQ. 1) GO TO 53
50    IF (BEGCYC .GE. TSU) GO TO 51
      D1 =(TSU - BEGCYC) * VELM
```

```
      D2 = TSN * VELSNM
      V1 = VELM
      V2 = VELSNM
      NCRK = 1
      T = TSU
      T1 = 0
      NCR = 0
      GO TO 52
51    D1 = (CYC - BEGCYC) * VELSNM
      D2 = TSU * VELM
      V1 = VELSNM
      V2 = VELM
      T = 0
      T1 = TSU
      NCR = 1
      NCRK = 0
52    DELDSN = VELSNM * STEP
      CYCD = TSU * VELM + TSN * VELSNM
53    DELTAD = VELM * STEP
      DO 130 N = 1,NB
      IF (NPLIND .EQ. 0) GO TO 60
      R = ANPTS - 1.0
      DISC = R**2 -(BS *(XS - XB(N)) - BC*(YS - YB(N)))**2
      IF (DISC .LT. 0.)GO TO 130
60    DO 100 J = 1,NFM
      IF (NMDET(J) .EQ. 1) GO TO 100
      ANFM = FMF(N) + FMD(J)
      IF (NPLIND .EQ. 0) GO TO 61
      NR = NPTS
63    IF (ANFM .GE. PLC(NR)) GO TO 62
      NR = NR - 1
      IF (NR .EQ. 0) GO TO 100
      GO TO 63
62    R = NR - 1
      R = R + ((ANFM - PLC(NR))/(PLC(NR+1) - PLC(NR)))
      GO TO 64
61    R = 1
      IF (ANFM .GT. 66.0) R = 10**((ANFM - 66.0)/17.0)
64    DISC = R**2 - (BS *(XS - XB(N)) - BC *(YS - YB(N)))**2
      IF (DISC .LT. 0.)GO TO 100
      DISC = SQRT(DISC)
      DLARGE = DISC - BC *(XS - XB(N)) - BS *(YS - YB(N))
      DSMALL = -DISC - BC *(XS - XB(N)) - BS *(YS - YB(N))
      IF (DSMALL .GE. 0.0) GO TO 70
      DSMALL = 0.0
      IF (DLARGE .LT. 0.0) GO TO 100
70    IF (DLARGE .GT. DIS) DLARGE = DIS
      IF (MODE .EQ. 1) GO TO 72
```

```
      C = DSMALL
      CSMALL = DSMALL - D1
      NCYC = CSMALL/CYCD
      P = NCYC * CYCD
      X = XS + BC *(D1 + P)
      Y = YS + BS *(D1 + P)
      CSMALL = DSMALL - P
      NEVN = (C1/V1 + NCYC * CYC)/STEP
      IF (DSMALL .GT. D2) GC TO 74
      V = V2
      TREM = CSMALL/V2
      GC TO 76
74    CSMALL = DSMALL - D2
      X = X + BC * D2
      Y = Y + BS * D2
      V = V1
      MC2 = (C2/V2)/STEP
      NEVN = NEVN + MD2
      TREM = CSMALL/V1
      T = T1
      NCRK = NOR
76    C = C - DSMALL
      NREM = TREM/STEP
      TEMP = NREM * ISTEP
      IF (TREM .GT. TEMP      ) NREM = NREM + 1
      NSEGS = NEVN + NREM
      GC TO 75
72    V = VELM
      C = 0.0
      X = XS
      Y = YS
75    TC = DSMALL/V
      K = TO/STEP
      TEMP = K * ISTEP
      IF (TO .NE. TEMP      ) TO = (K + 1)* ISTEP
      CSMALL = D + V * TC
      X = X + BC * V * TC
      Y = Y + BS * V * TC
      IF (MODE .EQ. 1) GC TC 80
      T = T + TO
      IF (T .GE. CYC) T = T - CYC
80    IF (CSMALL .GT. DLARGE) GO TO 1CC
      IF (MON(N) .EQ. 2) GO TC 82
      IF (MODE .EQ. 1) NSEGS = (DSMALL/V)/STEP
      IF (NOW(NSEGS,N) .EQ. 0) GO TC 9C
82    IF((MODE .EQ. 0).ANC.(NCRK .EQ. 0))GO TC 92
      C(N) = SQRT((XB(N) - X)**2 + (YB(N) - Y)**2)
      IF (NPLIND .EQ. 0) GO TO 84
```

```
      IF (C(N) .GE. ANPTS -1.) GO TO 90
      IC = C(N)
      AIC = IC
      PL(N) = PLC(IC+1) + (PLC(IC+2) - PLC(IC+1))*(C(N) - AIC)
      GO TO 88
84.   IF (C(N) .GT. 1.)GO TO 86
      PL(N) = 66.0
      GO TO 88
86    PL(N) = 17. * ALOG10(C(N)) + 66.
88    IF (PL(N) .GT. ANFM) GO TO 90
      NFD(J) = NFD(J) + 1
      NMDET(J) = 1
      AK1(J) = AK1(J) + 1.0/ANRUNS
      GO TO 100
90    IF (MODE .EQ. 1) GO TO 94
92    T = T + STEP
      NSEGS = NSEGS + 1
      IF (T .GT. TSU) GO TO 96
      NCRK = 0
94    X = X + DELTAX
      Y = Y + DELTAY
      DSMALL = DSMALL + DELTAD
      GO TO 80
96    IF (T .GE. CYC) T = 0
      NCRK = 1
      X = X + DELXSN
      Y = Y + DELYSN
      DSMALL = DSMALL + DELDSN
      GO TO 80
100   CONTINUE
      NSUM = 0
      DO 110 J = 1,NFM
110   NSUM = NMDET(J) + NSUM
      IF (NSUM .EQ. NFM) GO TO 140
130   CONTINUE
140   CONTINUE
      RETURN
      END
      SUBROUTINE TRSH
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SF1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
```

```
     *          NHT3(6),NF(6),          LSA(6),CD1(6)
C         NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C         MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JFLCT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C         AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C         AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C         AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,BUOYSP,JZONE0,JZONE1
      COMMON/A/ M,ISTEP,         J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMOT(6),
     2          NNFD(6), JJN
C
C
C
C
C
C     SUBROUTINE THRESHOLD/OPTIMIZATION
```

```
C          WHEN THIS SUBROUTINE IS CALLED DURING AREA MAXIMIZATION, THERE
C          WILL IN GENERAL BE SEVERAL F.M. SETS.  THE STATES OF EACH OF
C          THESE MUST BE INITIALIZED TO 0 EACH TIME THIS SUBROUTINE IS
C          CALLED.
1000  DO 1100 M=1,NFMS
      LIM(M) = .FALSE.
1100  LS(M) = 0
C
      M = 1
C
1080  LIM1 = .FALSE.
      IF (LPT .LT. 3) GO TO 1010
      IF (LAS(M) .GE. 5) GO TO 1210
C
C COMPLTE COORDINATES OF STARTING AND ENDING RECTANGLES
      IF (NHOWMV .EQ. 1) GO TO 1090
      XS2 = CCD(M)
      XS1 = -XS2
      XF2 = XS2
      XF1 = XS1
      YS1 = XS2 + DF
      YS2 = YS1
      YF1 = XS1
      YF2 = XS1
      GO TO 1010
C
1090  XS2 = CCD(M) - DF - DISMAX
      XS1 = -XS2
      YS1 = XS1
      YS2 = XS2
C
1010  LSM = LS(M)
      IF (LSM .GE. 8) GO TO 1210
      SPIM = SPI(M)
      IF((LSM .NE. 3).AND. .NOT. LIM(M)) GO TO 1015
      IF (ABS(SPIM) .GT. SPMIN) GO TO 1015
      LS(M) = 9
      GO TO 1210
C
1015  XSP = SP(M)
      IF (LSM .NE. 0) XSP = XSP + SPIM
C          NFM MUST BE FETCHED AT THIS POINT.  IT IS USED BY REFINE IN
C          CASE LSM = 3.
      NFM = NF(M)
      M1 = M
      IF (LSM - 3) 1031,1018,1040
1018  LT = 3
      WRITE (6,2) LT
```

```
2      FORMAT (4H P   ,5X,4HLT =, I5)
       CALL REFN
       WRITE (6,3) L, NA
3      FORMAT (4H Q   ,5X,3HL =, I5,3X,4HNA =, I5)
1020   IF (L .EQ. 0) GO TO 1030
       LS(M) = 9
       IF (NA .GT. 0) LS(NFMS) = LSM
       GO TO 1210
C
1030   IF (NA .GT. 0) LS(NFMS) = 9
       GO TO 1040
C
1031   IF (XSP .LT. SPMIN) GO TO 1032
       IF (XSP .GT. SPMAX) GO TO 1033
       GO TO 1040
C
1032   SPLIM = SPMIN
       GO TO 1034
1033   SPLIM = SPMAX
1034   IF ((LSM.LE.0).OR.(LSM.GE.3)) GO TO 241
       GO TO (1035,1037),LSM
C
C         LSM = 0
241    SP(M)=SPLIM
       XSP = SPLIM
       GO TO 1040
C
C         LSM = 1
1035   IF (LIM1) GO TO 1036
       SPI(M) = -SPIM
       LIM1 = .TRUE.
       GO TO 1010
1036   SPI(M) = .5*SPIM
       GO TO 1010
C
C         LSM = 2
1037   SPDIF = SPLIM - SP(M)
       SPIM = .5*SPIM
       LIM(M) = .TRUE.
       IF (ABS(SPDIF) .GE. ABS(SPIM)) GO TO 1038
       SPI(M) = -SPIM
       XSP = SP(M) - SPIM
       LSM = 5
       GO TO 1040
1038   SPI(M) = SPDIF
       XSP = SPLIM
       LSM = 6
C
```

```
C COMPUTE PROBABILITIES FOR SET M
1040  KK = JPM(M1,4)
      IND = JFMI(M,1)
      FM1 = FM(IND)
      IF (NFM .LT. 2) GO TO 1060
C COMPUTE F.M. DIFFERENCES FOR REMAINING F.M.'S IN SET M
      DO 1050 N=2,NFM
      IND = JFMI(M,N)
1050  FMD(N) = FM(IND) - FM1
1060  FMD(1) = 0.
C
C CALCULATE BUOY POSITIONS
      IND = IB1
      DO 1070 J=1,NB
      XB(J) = FLOAT(KXB(IND))*XSP
      YB(J) = FLOAT(KYB(IND))*XSP
1070  IND = IND + 1
C
C PERFORM PROBABILITY CALCULATIONS
      IF (ISPEED .EQ. 0) GO TO 100
      ISEED(M) = JJN
      CALL PROBF
      IF (ICFLG .LT. 0) RETURN
      DO 120 J = 1,NFM
      IND = JFMI(M,J)
120   P1(IND,KK) = AK1(J)
      GO TO 130
100   CALL PROB
      IF (ICFLG .LT. 0) RETURN
C SAVE VALUES
1110  DO 1120 J=1,NFM
      IND = JFMI(M,J)
      P1(IND,KK) = AK1(J)
      P2(IND,KK) = AK2(J)
      P3(IND,KK) = AK3(J)
      HT1(IND,KK) = HTM1(J)
      HT2(IND,KK) = HTM2(J)
      HT3(IND,KK) = HTM3(J)
1120  TD(IND,KK) = TDM(J)
C
C DO THRESHOLD CALCULATION
130   LT = 1
      CALL REFN
1130  IF (L .EQ. 0) GO TO 1140
C  L=1 IMPLIES THAT PROBABILITIES EXCEED THRESHOLD
      LS(M) = 8
      SP(M) = XSP
      IF (NA .EQ. 0) GO TO 1210
```

```
      M1 = NFMS
      NFM = NA
      GO TO 1170
C
1140  IF (NA .EQ. 0) GO TO 1170
      LS(NFMS) = 8
      SF(NFMS) = XSP
C
1170  IF (LSM .EQ. 0) GO TO 1190
C DC COMPARISON TEST
      LT = 2
      CALL REFN
C UPDATE STATE OF SET M1
1180  MM = M1
      IR = 2*LSM - L
      ASSIGN 1185 TO IUPD
      GO TO 3000
1185  IF (NA .EQ. 0) GO TO 1200
C UPDATE STATE OF SET NFMS
      MM = NFMS
      IR = 2*LSM + L - 1
      ASSIGN 1200 TO IUPD
      GO TO 3000
C
C UPDATE FOR STATE 0 -- COLD START
1190  JPM(M1,4) = JPM(M1,2)
      JPM(M1,2) = KK
      LS(M1) = 1
C
1200  CONTINUE
      GO TO 1010
1210  M = M + 1
      IF (M .LE. NFMS) GO TO 1080
      RETURN
C
C
C
C     SUBROUTINE UPDATE
3000  K = JPM(MM,4)
      IF((IR.LT.1).OR.(IR.GT.14)) GO TO 3020
      GO TO (3030,3020,3030,3040,3050,3060,3070,3080,3070,3020,3083,
     * 3020,3070,3087), IR
C
3020  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = K
      SPI(MM) = -SPI(MM)
      LS(MM) = 2
      GO TO 3090
```

```
3030  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = JPM(MM,2)
      JPM(MM,2) = K
      SP(MM) = SP(MM) + SPI(MM)
      LS(MM) = 2
      GO TO 3090
3040  JPM(MM,4) = JPM(MM,3)
      JPM(MM,3) = K
      SPI(MM) = .5*SPI(MM)
      LS(MM) = 3
      GO TO 3090
3050  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = JPM(MM,2)
      JPM(MM,2) = K
      SP(MM) = SP(MM) + SPI(MM)
      SPI(MM) = .5*SPI(MM)
C NO CHANGE IN STATE
      GO TO 3090
3060  JPM(MM,4) = JPM(MM,3)
      JPM(MM,3) = K
      SPI(MM) = -SPI(MM)
      LS(MM) = 4
      GO TO 3090
3070  JPM(MM,4) = JPM(MM,3)
      JPM(MM,3) = JPM(MM,2)
      JPM(MM,2) = K
      SP(MM) = SP(MM) + SPI(MM)
      SPI(MM) = -.5*SPI(MM)
      LS(MM) = 3
      GO TO 3090
3080  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = K
      SPI(MM) = -.5*SPI(MM)
      LS(MM) = 3
      GO TO 3090
3083  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = JPM(MM,2)
      JPM(MM,2) = K
      SP(MM) = SP(MM) + SPI(MM)
      SPI(MM) = -.5*SPI(MM)
      LS(MM) = 7
      GO TO 3090
3087  JPM(MM,4) = JPM(MM,1)
      JPM(MM,1) = K
      SPI(MM) = .5*SPI(MM)
C NO CHANGE IN STATE
3090  GO TO ILPD, (1185, 1200)
      END
```

```
      SUBROUTINE REFN
C     SUBROUTINE REFINE
C
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),FL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),        LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SFGRT(40),SFIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JFLCT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,FSCIF,CDMIN,CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,BUOYSP,JZONEO,JZONE1
      COMMON/A/ M,ISTEP,        J,K
      COMMON/A/ IGFLG,IGFLG
```

```
      LOGICAL    GECEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GECEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2          NNFD(6), JJN
C
C
2000  N = 1
      N1 = 1
      NA = 0
      ASSIGN 2100 TO ITESTS
      GO TO 5000
2100  N = 2
      L = LD
C
2010  IF (N.GT.NFM) GO TO 2040
      ASSIGN 2015 TO ITESTS
      GO TO 5000
2015  IF (L.EQ.LD) GO TO 2020
      NA = NA + 1
      JFMI(NFMS+1,NA) = JFMI(M1,N)
      GO TO 2030
C
2020  N1 = N1 + 1
      JFMI(M1,N1) = JFMI(M1,N)
C
2030  N = N + 1
      GO TO 2010
C
2040  IF (NA .EQ. 0) GO TO 2060
      NFMS = NFMS+1
      NF(NFMS) = NA
      NF(M1) = N1
      NFM = N1
      SP(NFMS) = SP(M1)
      SPI(NFMS) = SPI(M1)
      SPO(NFMS) = SPO(M1)
      SP1(NFMS) = SP1(M1)
      CD(NFMS) = CD(M1)
      CD1(NFMS) = CD1(M)
      CDI(NFMS) = CDI(M1)
      LAS(NFMS) = LAS(M1)
      LFA(NFMS) = LFA(M1)
      CDD(NFMS) = CDD(M1)
      LIM(NFMS) = LIM(M1)
      IGEA(NFMS) = IGEA(M1)
      DO 2050 N=1,6
```

```
2050  JPM(NFMS,N) = JPM(M1,N)
2060  RETURN
C
C
C
C        SUBROUTINE TESTS, N IS INDEX IN CURRENT F.M. SET--NO. M
5000  LD = 0
      IF((LT .LT. 0).OR.(LT .GT. 3))GO TO 5020
      GO TO (5010, 5020, 5030), LT
C
C     LT = 1, THRESHOLD TEST
5010  IF (AK1(N) .GT. THRESH) LD = 1
      GO TO 5070
C
C        LT = 2 OR LT .GT. 3, COMPARISON TESTS
C        KK HAS INITIALLY BEEN SET TO JPM(M1,4) (FOR THRESHOLD/OPTIMUM)
C        OR TO JPM(M1,L), L = 5 OR 6 (FOR AREA MAXIMIZATION)
5020  K = JPM(M1,LT)
      J = JFMI(M1,N)
      IF (P1(J,KK) .GT. P1(J,K)) LD = 1
      GO TO 5070
C
C        LT = 3, OPTIMUM TEST
5030  J = JFMI(M1,N)
      K = JPM(M1,1)
      D = P1(J,K)
      K = JPM(M1,2)
      E = P1(J,K)
      K = JPM(M1,3)
      F = P1(J,K)
      IF(((D-F)**2) .LE. (PSDIF*(2.*E -D -F))) LD = 1
C
5070  GO TO ITESTS, (2100, 2015)
      END
      SUBROUTINE GEOM
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),         LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
```

```
C         MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C         AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C         AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C         AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,FSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNC,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHOWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,BUOYSP,JZONE0,JZONE1
      COMMON/A/ M,ISTEP,      J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2          NNFD(6), JJN
C
C
C
C         GET-GEOMETRY-DATA SUBROUTINE, IF GEOEND='TRUE' AT RETURN,
C         NO ADDITIONAL GEOMETRIES ARE AVAILABLE
      IF (IGFLG .EQ. 2) GO TO 6000
 4000 IF (KN .NE. 0) GO TO 4110
 4100 GEOEND = MINB .GT. MAXB
      IF (GEOEND) GO TO 4060
      N = MINO(NHOWMV,2)
```

```
      I1 = KGC(MAXB, MODE+1, N)
      KN = KGC(MAXB+1, MODE+1, N) - I1
      MAXB = MAXB - 1
      IF (KN .EQ. 0) GO TO 4100
4110  IGE = KGI(I1)
      I1 = I1 + 1
      KN = KN - 1
C
C     COLLECT GEOMETRY DATA
C
6000  IB1 = JBUOY(IGE)
      NB = JBUOY(IGE + 1) - IB1
      BUOYSP = (KXB(IB1) - KXB(IB1+1))**2 + (KYB(IB1) - KYB(IB1+1))**2
      BUOYSP = SQRT(BUOYSP)
      SPMAX = RFRNGE / RADIUS(IGE)
      SPMIN = SPGRAN / BUOYSP
      MN = 2
      IF (NCHAN .GE. NB) GO TO 4120
      MN = 0
      TMON = TMN(IGE, ICHAN)
      JZONE0 = JZONE(IGE, ICHAN)
      JZONE1 = JZONE(IGE + 1, ICHAN) - 1
4120  DO 4010 J = 1,NB
4010  MON(J) = MN
      IF (MN .EQ. 2) GO TO 4060
      TREAL = 0.
      JZ = JZONE0
C
      DO 4050 J=1,NSEGMX
      TREAL = TREAL + STEP
      D = TREAL - TMON
      IF (D .LT. 0.) GO TO 4030
      TREAL = D
      IF (JZ .EQ. JZONE1) GO TO 4020
      JZ = JZ + 1
      GO TO 4030
4020  JZ = JZONE0
4030  DO 4040 K=1,NB
4040  NCW(J,K) = 0
      DO 4050 K=1,NCHAN
      GO TO (4042, 4045, 4048), ICHAN
4042  KX = IZONE4(K,JZ)
      GO TO 4050
4045  KX = IZONE8(K,JZ)
      GO TO 4050
4048  KX = IZONEC(K,JZ)
4050  NCW(J,KX) = 1
C
```

```
 406C  RETURN
       END
       SUBROUTINE OPUT
C
C         MAX. NO. OF BUOYS IS 16
       COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1           SE(6,16),MON(16)
C         MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
       COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *           INDALT(6),FM(6),TOM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *           SPI(6),SPF(6),SPO(6),SF1(6),JFMI(6,6),CD(6),CDI(6),
     *           CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *           NHT3(6),NF(6),       LSA(6),CD1(6)
C         NUMBER OF POSITION INDEXES IS 6
       COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *           HT3(6,6),TD(6,6)
C         MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
       COMMON/A/ PLC(205)
       COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
       COMMON/A/ IDT(12)
       COMMON/A/ KGC(17,2,2)
C         MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
       COMMON/A/ KGI(99)
C         MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
       COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
       COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1           SPIGRH(40),JZONE(41,3),JBUOY(41),JFLCT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
       COMMON/A/ KXB(480),KYB(480)
C         AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
       COMMON/A/ IZONE4(4,200)
C         AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
       COMMON/A/ IZONE8(8,120)
C         AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
       COMMON/A/ IZONEC(12,100)
       COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *           KON1,KON2,KON3,SPGRAN,FSDIF,CDMIN,CDMAX,CDC,CDIO,
     *           NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
       COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *           MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *           DIRMIN,DIRMAX,DISMAX,NFOWMV,NPLINC,NSEGMX,FFRNGE,THRESH,
     *           TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *           IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,BUOYSP,JZONEC,JZONE1
       COMMON/A/ M,ISTEP,         J,K
       COMMON/A/ ICFLG,IGFLG
```

```
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMOT(6),
     2          NNFD(6), JJN
C
C
C
C     SUBROUTINE TO PRINT 1ST PART
2004  FORMAT (12H1GEOMETRY ID, 4A4)
2005  FORMAT (30X, 21HSPACING (NAUT. MI.) =,1X,5(F5.0,7X),F5.0)
2006  FORMAT (20X, 31HMEAN TIME TO DETECTION (MIN.) =,1X,5(F5.0,1X,A4,
     * 1X),F5.0,1X,A4)
2007  FORMAT (21H MEAN HOLDING TIME OF, I2, 28H OR MORE DETECTIONS (MIN.
     *) =,1X,5(F5.0,7X),F5.0)
2013  FORMAT (3X, I5, 12I5)
2016  FORMAT(30X,21HFIGURE OF MERIT(DB) =,  1X,5(F7.2,3X),F7.2)
2018  FORMAT (17H NUMBER OF BUOYS=,I5/)
2021  FORMAT (1X,46HEFFECTIVE RADIUS OF AREA COVERED (NAUT. MI.) =,1X,
     1      6F11.2)
2025  FORMAT (//, 26H MONITORING PERIOD (MIN.)=, F6.1/)
2026  FORMAT (//,100H ABBREVEATIONS--ND = NO DETECTION, LC = LOW CONFIDE
     1NCE, MC = MEDIUM CONFIDENCE, HC = HIGH CONFIDENCE, ///)
2027  FORMAT(1X,14HPROBABILITY OF,I2,1X,33HOR MORE SIMULTANEOUS DETECTIO
     2NS =,1X,5(F7.2,5X),F7.2)
2036  FORMAT (22H CONTINUOUS MONITORING)
      IF (IOFLG .EQ. 3) GO TO 8000
7000  WRITE (6,2004) (IDENT(I,IGE),I=1,4)
      WRITE (6,2018) NB
      IF (MN .NE. 2) GO TO 265
      WRITE (6,2036)
      GO TO 295
265   WRITE (6,2025) TMON
      DO 294 K=JZONE0,JZONE1
      GO TO (270,280,290), ICHAN
270   DO 273 J=1,4
273   JZONET(J)=IZONE4(J,K)
      J = 4
      GO TO 260
280   DO 283 J=1,8
283   JZONET(J) = IZONE8(J,K)
      J = 8
      GO TO 260
290   DO 293 J=1,12
293   JZONET(J) = IZONEC(J,K)
      J = 12
260   WRITE (6,2013) J, (JZONET(I), I = 1,12)
```

```
294   CONTINUE
295   CONTINUE
C
C
C
C     SUBROUTINE TO PRINT 2ND PART
      IF (LPT .EQ. 3) GO TO 7250
      WRITE (6,2016) (FM(I),I = 1,MFM)
      GO TO 7500
7250  WRITE (6,2016) (FMCT(I),I = 1,MFM)
      WRITE (6,2021) (CDD(I),I = 1,MFM)
7500  WRITE (6,2005) (BSPF(I), I = 1,MFM)
      WRITE (6,2027) KON1, (BAK1(I), I = 1,MFM)
      WRITE (6,2027) KON2, (BAK2(I), I = 1,MFM)
      WRITE (6,2027) KON3, (BAK3(I), I = 1,MFM)
      WRITE (6,2006) (BTDM(I),NNFD(I), I = 1,MFM)
      WRITE (6,2007) KON1, (BHTM1(I), I = 1,MFM)
      WRITE (6,2007) KON2, (BHTM2(I), I = 1,MFM)
      WRITE (6,2007) KON3, (BHTM3(I), I = 1,MFM)
      WRITE(6,2026)
      RETURN
C
C
C
C     SUBROUTINE TO PREPARE DATA FOR PRINTING
8000  IF (IOUT .EQ. 0) GO TO 8005
8050  JJN = ISEED(M)
      WRITE (6,2) JJN
2     FORMAT (6H JJN =,I10)
      JSAVE = J
      KSAVE = K
      LSAVE = L
      NSAVE = N
      IND = JFMI(M,1)
      FM1 = FM(IND)
      IF (NFM .LT. 2) GO TO 8070
      DO 8060 I = 2,NFM
      IND = JFMI(M,I)
8060  FMD(I) = FM(IND) - FM1
8070  FMD(1) = 0.0
      IND = IB1
      DO 8080 I = 1,NB
      XB(I) = FLOAT(KXB(IND))*SP(M)
      YB(I) = FLOAT(KYB(IND))*SP(M)
8080  IND = IND + 1
      IF (LPT .LT. 3) GO TO 8085
      IF (NHOWMV .EQ. 1) GO TO 8083
      XS2 = CDD(M)
```

```
      XS1 = -XS2
      XF2 = XS2
      XF1 = XS1
      YS1 = XS2 + DF
      YS2 = YS1
      YF1 = XS1
      YF2 = XS1
      GO TO 8085
8083  XS2 = CCC(M) - DF - DISMAX
      XS1 = -XS2
      YS1 = XS1
      YS2 = XS2
      CALL PROB
      J = JSAVE
      K = KSAVE
      L = LSAVE
      N = NSAVE
      DO 8090 I = 1,NFM
      IND = JFMI(M,I)
      P1( IND,K) =  AK1(I)
      P2( IND,K) =  AK2(I)
      P3( IND,K) =  AK3(I)
      HT1(IND,K) = HTM1(I)
      HT2(IND,K) = HTM2(I)
      HT3(IND,K) = HTM3(I)
8090  TD( IND,K) =  TDM(I)
      ICUT = 0
      ITM = M
8085  BSPF(L) = BUCYSP*SF(M)
      BTDM(L) = TD(J,K)
      BHTM1(L) = HT1(J,K)
      BHTM2(L) = HT2(J,K)
      BHTM3(L) = HT3(J,K)
      BAK1(L) = P1(J,K)
      BAK2(L) = P2(J,K)
      BAK3(L) = P3(J,K)
      IF (BTDM(L) .GT. 0.) GO TO 8010
      NNFD(L) = NDET
      BTDM(L) = 9999.
      GO TO 8040
8010  IF (BAK1(L) .GT. T1) GO TO 8020
      NNFD(L) = LCON
      GO TO 8040
8020  IF (BAK1(L) .GT. T2) GO TO 8030
      NNFD(L) = MCON
      GO TO 8040
8030  NNFD(L) = ICON
8040  RETURN
```

```
      END
      SUBROUTINE AMAX
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MCN(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),        LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,FSCIF,CDMIN,CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MCCE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
     *          DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *          IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,BUOYSP,JZONEO,JZONE1
      COMMON/A/ M,ISTEP,      J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
```

```
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTOM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1           BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2           NNFD(6), JJN
      DIMENSION IFM(6)
      EQUIVALENCE (LS(1),IFM(1))
C
C
C
C        AREA MAXIMIZATION PROBLEM
9000  CD(1) = CDO
      NFMS = 1
      LAS(1) = 0
      LFA(1) = .FALSE.
      IGEA(1) = 0
      DO 9003 N=1,NFM
9003  JFMI(1,N) = N
      NF(1) = NFM
      DO 9007 K=1,6
9007  JFM(1,K) = K
      IF (NHOWMV .GT. 1) GO TO 9010
      CDMIN = AMIN1(CDMIN, DF + DISMAX)
      DIRMIN = 0.
      DIRMAX = 360.
C
9010  IOFLG = 1
      CALL GEOM
9015  IF (GEOEND) GO TO 9220
      IF (NHOWMV .GE. 2) GO TO 9017
      SPR = SPGRH(IGE)
      SPIR = SPIGRH(IGE)
      GO TO 9018
9017  SPR = SPGRT(IGE)
      SPIR = SPIGRT(IGE)
9018  DO 9020 M=1,NFMS
      IF (LAS(M) .EQ. 8) GO TO 9020
      LAS(M) = 0
      SP(M) = SPR*CD(M)
      SPI(M) = SPIR*CD(M)
      CDD(M) = CD(M)
      CDI(M) = CDIO
9020  CONTINUE
C
9030  CALL TRSH
      WRITE (6,5) IOFLG
5     FORMAT (4H AH ,5X,7HIOFLG =, I5)
      IF (IOFLG .LT. 0) RETURN
```

```
C
9040  NFMS1 = NFMS
      M = 1
9045  LSA(M) = 2
      LASM = LAS(M)
      LSM = LS(M)
      NFM = NF(M)
      M1 = M
      KH = JPM(M1,5)
      LT = 2
      J = 2*LASM - LSM + 10
      IF((J.LT.1) .OR. (J.GT.10)) GO TO 9046
      GO TO (9050,9070,9080,9130,9140,9150,9160,9180,9200,9210), J
9046  LSA(M) = 0
      GO TO 9220
9050  IF (.NOT. LFA(M)) GO TO 9060
      LAS(M) = 3
      GO TO 9220
9060  CDI(M) = -CDI(M)
      LAS(M) = 1
      GO TO 9220
9070  LAS(M) = 4
      GO TO 9215
9080  CALL REFN
C        IF L=1, THE OLD VALUE IS BIGGER THAN THE NEW VALUE
9090  IF (L .EQ. 1) GO TO 9100
      IF (NA .EQ. 0) GO TO 9110
      M1 = NFMS
9100  CDI(M1) = -CDI(M1)
      CD(M1) = CD(M1) + CDI(M1)
      LAS(M1) = 3
      LSA(M1) = 0
      M1 = M
9110  IF (L .EQ. 0) GO TO 9120
      IF (NA .EQ. 0) GO TO 9220
      M1 = NFMS
9120  LAS(M1) = 3
      LSA(M1) = 2
      IF (LFA(M1)) LAS(M1) = 5
      GO TO 9220
9130  LAS(M) = 2
      CDI(M) = -CDI(M)
      SP1(M) = SP(M)
      GO TO 9215
9140  SPC(M) = SP(M)
      SPI(M) = 4.*SPI(M)
      LSA(M) = 0
      GO TO 9220
```

```
9150  CD(M) = CDD(M)
      SP1(M) = SP(M)
      GO TO 9215
9160  CALL REFN
C        IF L=1, THE OLD VALUE IS BIGGER THAN THE NEW VALUE
9170  IF (L .EQ. 1) GO TO 9173
      IF (LFA(M)) LAS(M) = 5
      IF (NA .EQ. 0) GO TO 9220
      LAS(NFMS) = 5
      LSA(NFMS) = 0
      CD(NFMS) = CD(NFMS) - CDI(NFMS)
      GO TO 9220
9173  LAS(M) = 5
      LSA(M) = 0
      CD(M) = CD(M) - CDI(M)
      IF (NA .EQ. 0) GO TO 9220
      LSA(NFMS) = 2
      IF (LFA(M)) LAS(NFMS) = 5
      GO TO 9220
9180  IF (CDI(M)  .GT. 0.) GO TO 9190
      CDI(M) = -CDI(M)
      LAS(M) = 2
      GO TO 9215
9190  LAS(M) = 4
      GO TO 9215
9200  LAS(M) = 2
      CD(M) = CD(M) - CDI(M)
      SPC(M) = SP(M)
      SPI(M) = 4.*SPI(M)
      LSA(M) = 0
      GO TO 9220
9210  IF (.NOT. LFA(M)) GO TO 9215
      LAS(M) = 8
      IGEA(M) = IGE
      CD1(M) = CD(M)
C
9215  LSA(M) = 4
      SP1(M) = SP(M)
9220  M = M + 1
      IF (M .LE. NFMS1) GO TO 9045
C
      AC = .FALSE.
      M = 1
9225  LASM = LAS(M)
      L = 5
      IF ((LASM.LE.0).OR.(LASM.GE.6)) GO TO 9226
      GO TO (9230,9270,9230,9230,9283), LASM
9226  L=6
```

```
      GO TO 9300
C
9230  AC = .TRUE.
      SPI(M) = 4.*SPI(M)
      CDD(M) = CD(M) + CDI(M)
      LFA(M) = .FALSE.
      IF (CDD(M) .GE. CDMAX) GO TO 9240
      IF (CDD(M) .GT. CDMIN) GO TO 9260
      CDLIM = CDMIN
      GO TO 9250
9240  CDLIM = CDMAX
9250  CDI(M) = CDLIM - CD(M)
      CDD(M) = CDLIM
      LFA(M) = .TRUE.
9260  CD(M) = CDD(M)
      IF (LSA(M) .NE. 0) SPO(M) = SP(M)
      SP(M) = SP(M) + SPR*CDI(M)
      GO TO 9300
C
9270  IF (CDI(M) .LE. CDMIN) GO TO 9280
      CDI(M) = .5*CDI(M)
      CDD(M) = CD(M) + CDI(M)
      SP(M) = .5*(SPO(M) + SP1(M))
      AC = .TRUE.
      GO TO 9300
9280  IGEA(M) = IGE
      CD1(M) = CD(M)
      LFA(M) = .TRUE.
      LAS(M) = 7
      IF (LSA(M) .EQ. 0) LSA(M) = 5
      L = 6
      GO TO 9300
C
9283  M1 = M
      LAS(M1) = 7
      LT = LSA(M1)
      IF (LT .EQ. 0) LT = 5
      LSA(M1) = LT
      IF (IGEA(M1)) 9286, 9295, 9310
9286  KK = JPM(M1,6)
      NFM = NF(M1)
      CALL REFN
C        IF L = 1, THE VALUE FROM A PREVIOUS GEOMETRY IS BIGGER
9289  IF (L .EQ. 1) GO TO 9292
      IF (NA .EQ. 0) GO TO 9295
      LAS(NFMS) = 7
      LSA(NFMS) = 0
      GO TO 9295
```

```
9292  LSA(M1) = 0
      IF (NA .EQ. 0) GO TO 9310
      LAS(NFMS) = 7
      LSA(NFMS) = LT
      IGEA(NFMS) = -IGE
      M1 = NFMS
C
9295  IGEA(M1) = -IGE
      CD1(M1) = CD(M1)
      IF (LT .EQ. 2) GO TO 9298
      SP1(M1) = SPC(M1)
      GO TO 9299
9298  SP1(M1) = SP(M1)
9299  L = 6
C
9300  J = LSA(M)
      IF (J .EQ. 0) GO TO 9310
C         SAVE VALUES
      K = JPM(M,L)
      JPM(M,L) = JPM(M,J)
      JPM(M,J) = K
C
9310  M = M + 1
      IF (M .LE. NFMS) GO TO 9225
C
      IF (AC) GO TO 9030
      GO TO 9010
C
9320  KK = NFM
C
      DO 9380 M1=1,NFMS
      IGE = IABS(IGEA(M1))
      IF (IGE .EQ. 0) GO TO 9380
      DO 9330 N=1,KK
9330  IFM(N) = 0
C
      DO 9350 M=M1,NFMS
      N = IABS(IGEA(M))
      IF (N .NE. IGE) GO TO 9350
      IGEA(M) = 0
      SP(M) = SP1(M)
      NFM = NF(M)
      DO 9340 N=1,NFM
      J = JFMI(M,N)
9340  IFM(J) = M
9350  CONTINUE
C
      IGFLG = 2
```

```
9320  KK = MFM
C
      DO 9380 M1=1,NFMS
      IGE = IABS(IGEA(M1))
      IF (IGE .EQ. 0) GO TO 9380
      DO 9330 N=1,KK
9330  IFM(N) = 0
C
      DO 9350 M=M1,NFMS
      N = IABS(IGEA(M))
      IF (N .NE. IGE) GO TO 9350
      IGEA(M) = 0
      SP(M) = SP1(M)
      NFM = NF(M)
      DO 9340 N=1,NFM
      J = JFMI(M,N)
9340  IFM(J) = M
9350  CONTINUE
C
      IGFLG = 2
      CALL GEOM
9355  L = 0
      ICUT = 0
      IF (ISPEED .EQ. 1) IOUT = 1
      DO 9360 J=1,KK
      M = IFM(J)
      IF (M .EQ. 0) GO TO 9360
      L = L + 1
      NFM = NF(M)
      IF (M .NE. ITM) ICUT = 1
      FMCT(L) = FM(J)
      CDD(L) = CD1(M)
      K = JPM(M,6)
      ICFLG = 3
      CALL OPUT
9360  CONTINUE
C
      MFM = L
      ICFLG = 1
      CALL OPUT
9380  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RADN(RNND,JNJ)
      COMMON/B/ JCOUNT, KKK, MYSEED
      JCOUNT = JCOUNT + 1
      JNJ = JNJ * KKK
      RNND = 0.5 + FLOAT(JNJ) * 2.328306E-10
      RETURN
      END


      SUBROUTINE INDAT
C
C    SUBROUTINE TO READ GEOMETRY DATA INTO PROGRAM
C
C         MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C         MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SP(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),        LSA(6),CD1(6)
C         NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C         MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C         AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C         AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C         AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
```

```
     *           KON1,KON2,KCN3,SPGRAN,PSCIF,CDMIN,CCMAX,CDO,CDIO,              040
     *           NRUNS,JJNO,NDET,LCON,MCCN,ICON,T1,T2                           041
      CCMMON/A/  L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,      042
     *           MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,            043
     *           DIRMIN,DIRMAX,DISMAX,NHCWMV,NPLIND,NSEGMX,RFRNGE,THRESH,      044
     *           TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,        C45
     *           IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMCN,BUCYSP,JZCNEO,JZONE1         046
      CCMMON/A/  M,ISTEP,       J,K                                             047
      CCMMON/A/  IOFLG,IGFLG                                                    048
      LCGICAL    GEOEND,AC,HOLD1,HOLD2,HCLD3,LIM,LIM1,LFA                       049
      CCMMON/A/  GEOEND,AC,HOLD1(6),HOLD2(6),HOLC3(6),LIM(6),LIM1               050
      CCMMON/A/  LFA(6)                                                         C51
      CCMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),        052
     1           BAK2(6), BAK3(6),ISEEC(6), NPTS, ISPEED, IOUT, FMOT(6),        C53
     2           NNFD(6), JJN                                                   054
      DIMENSICN KG(4), KGS(4,40,16), KXBT(16), KYBT(16),NGE(40,4)
C     NUMBER OF GEOMETRIES IS  10
      DATA NGMAX/10/
998   FCRMAT ('1GEOMETRY NO.', I3/' GECMETRY ID ', 4A4/' NUMBER OF BUOYSTES00510
     *',I3//' TARGET APPLICATIONS')                                       TES00520
997   FORMAT (5X,4A4)                                                     TES00530
996   FCRMAT ('0',7X,'BUOY  DATA'/'OBUOY NO.    X       Y'/(I5,I9,I8))    TES00540
995   FORMAT ('OMONITORING PERIOD (MIN.) =', F6.1)                        TES00550
994   FCRMAT (3(I3, ','), I3)                                             TES00560
993   FCRMAT (7(I3, ','), I3)                                             TES00570
992   FORMAT (11(I3, ','), I3)                                            TES00580
985   FCRMAT ('ORADIUS =',F9.2/)                                          TES00650
1000  FCRMAT (4A4, 4I1, 5I5, 3F8.2/ 5F8.6)                                TES00670
1001  FCRMAT (16I5)
1002  FCRMAT(80A1)
      IGE = 1                                                             TES00900
      DC 10 I = 1,4                                                       TES00960
      DC 10 NB = 1, 16                                                    TES00970
10    NGE(NB, I) = 0                                                      TES00980
      JZA4 = 1                                                            TES00990
      JZA8 = 1                                                            TES01000
      JZAC = 1                                                            TES01010
      JBM = 1                                                             TES01020
      JP = 1                                                              TES01030
C                                                                         TES01040
90    READ (5, 1000, END=220) (IDENT(J,IGE),J=1,4), (KG(I), I=1,4),       TES01050
     *NB, JZ4, JZ8, JZC, NP, (TMN(IGE,I),I = 1,3),
     *SPGRT(IGE), SPIGRT(IGE), SPGRH(IGE), SPIGRH(IGE), RADIUS(IGE)
      WRITE(6,998) IGE, (IDENT(J,IGE), J=1,4), NB                         TES01100
      DC 100  I = 1,4                                                     TES01110
      IF (KG(I) .NE. 1) GO TO 100                                         TES01120
      J = NGE(NB, I) + 1                                                  TES01130
      KGS(I, J, NB) = IGE                                                 TES01140
```

```
      NGE(NB,I) = J                                                     TES01150
100   CCNTINUE                                                          TES01170
C                                                                       TES01180
      JZCNE(IGE,1) = JZA4                                               TES01190
      JZCNE(IGE,2) = JZA8                                               TES01200
      JZCNE(IGE,3) = JZAC                                               TES01210
      JBUOY(IGE) = JBM                                                  TES01220
      JPLOT(IGE) = JP                                                   TES01230
C                                                                       TES01240
      READ(5, 1001, END=210) (KXBT(I), KYBT(I), I=1,NB)                 TES01250
      WRITE(6,996) (I, KXBT(I), KYBT(I), I=1,NB)                        TES01260
      J = 1                                                             TES01270
      LIMT= JBM + NB - 1                                                TES01280
      DC 105 I=JBM,LIMT                                                 TES01300
      KXB(I) = KXBT(J)                                                  TES01310
      KYB(I) = KYBT(J)                                                  TES01320
105   J = J + 1                                                         TES01330
      JBM = LIMT+ 1                                                     TES01340
C                                                                       TES01350
      IF (NB .LE. 4) GO TO 115                                          TES01360
      WRITE(6,995) TMN(IGE,1)                                           TES01380
      LIMT= JZA4 + JZ4 - 1                                              TES01400
      READ(5, 1001, END=210) ((IZCNE4(I,J), I=1,4), J=JZA4,LIMT)        TES01420
      WRITE(6,994) ((IZONE4(I,J), I=1,4), J=JZA4,LIMT)                  TES01430
      IF (NB .LE. 8) GO TO 120                                          TES01460
      WRITE(6,995) TMN(IGE,2)                                           TES01480
      LIMT= JZA8 + JZ8 - 1                                              TES01500
      READ(5, 1CO1, END=210) ((IZONE8(I,J), I=1,8), J=JZA8,LIMT)        TES01520
      WRITE(6,993) ((IZONE8(I,J), I=1,8), J=JZA8,LIMT)                  TES01530
      JZA8 = LIMT+ 1                                                    TES01540
      IF (NB .LE. 12) GO TO 125                                         TES01560
      WRITE(6,995) TMN(IGE,3)                                           TES01580
      LIMT= JZAC + JZC - 1                                              TES01600
      READ(5, 1CO1, END=210) ((IZONEC(I,J), I=1,12), J=JZAC,LIMT)       TES01620
      WRITE(6,992) ((IZONEC(I,J), I=1,12), J=JZAC,LIMT)                 TES01630
      JZAC = LIMT+ 1                                                    TES01640
      GO TO 130                                                         TES01650
115   TMN(IGE,1) = 0.                                                   TES01670
120   TMN(IGE,2) = 0.                                                   TES01680
125   TMN(IGE,3) = 0.                                                   TES01690
C                                                                       TES01700
C                                                                       TES01710
C                                                                       TES02220
130   IF (KG(2) + KG(4) .NE. 0) GO TO 1410
      SPGRT(IGE) = 0.0
      SPIGRT(IGE) = 0.0
1410  IF (KG(1) + KG(3) .NE. 0) GO TO 1420                              TES02330
      SPGRH(IGE) = 0.0
```

```
      SPIGRH(IGE) = 0.0
1420  WRITE(6,985) RADIUS(IGE)
      JP = LIMT+ 1
C
C
      IGE = IGE + 1
      IF (IGE .LE. NGMAX) GO TO 90
C
      JZCNE(IGE,1) = JZA4
      JZCNE(IGE,2) = JZA8
      JZCNE(IGE,3) = JZAC
      JBUOY(IGE) = JBM
      JPLOT(IGE) = JP
C
      K = 1
      L = 1
      DC 260 I=1,2
      DC 250 J=1,2
      DC 240 NB=1,16
      KGC(NB, I, J) = L
      N = NGE(NB,K)
      IF (N .EQ. 0) GO TO 240
      L1 = L
      DC 230 M=1,N
      KGI(L) = KGS(K, M, NB)
      L = L + 1
230   CONTINUE
C
      LIMT= L - 1
240   CCNTINUE
C
      KGC(17, I, J) = L
250   K = K + 1
C
260   CCNTINUE
C
      ICE = IGE - 1
C
      JZA4 = JZA4 - 1
      JZA8 = JZA8 - 1
      JZAC = JZAC - 1
      JBM = JBM - 1
      JP = JP - 1
      L = L - 1
C
210   CCNTINUE
220   RETURN
      END
```

```
TES02460
TES02470
TES02480
TES02490
TES02500
TES02510
TES02900
TES02910
TES02920
TES02930
TES02940
TES02950
TES02960
TES02970
TES02990
TES03000
TES03020
TES03030
TES03040
TES03050
TES03060
TES03070
TES03080
TES03090
TES03110
TES03120
TES03130
TES03150
TES03160
TES03170
TES03180
TES03190
TES03200
TES03210
TES03230
TES03340
TES03350
TES03360
TES03370
TES03380
TES03390
TES03400
TES03410
```

COMPUTER PROGRAM LISTING B

TASDA AC PROGRAM

```
      BLOCK DATA
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/  XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *           SE(16)
      COMMON/A/  II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CCD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CC1
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/  JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C        MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/  PLC(205)
      COMMON/A/  JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/  IDT(12)
      COMMON/A/  KGC(17,2,2)
C        MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/  KGI(99)
C        MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/  IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/  RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *           SPIGRH(40),JBUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/  KXB(480),KYB(480)
      COMMON/A/  NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *           STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
     *           CDMAX,CDO,CDIC,
     *           NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/  L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *           YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *           ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *           NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *           IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/  M,ISTEP,       J,K
      COMMON/A/  IOFLG,IGFLG
      LOGICAL    GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/  GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/  BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *           BAK3,ISEED,NPTS,ICUT,FMCT,NNFD,JJN
      COMMON/B/  JCOUNT,KKK,MYSEED,NBSEED,JJJ,NBMAX
      COMMON/C/  NSTEP
C
C        MAXIMUM NO. OF SEGMENTS PER RUN IS 400
      DATA NSGMAX /400/
C        TASDA  BUILT-IN CONSTANTS
C        SIGR,SIGB ARE TEMPORAL AND SPATIAL SIGMAS FOR RANDO
C        PSDIF = 8*(SMALLEST VALUE OF DIFFERENCE BETWEEN EST
C                   MAXIMUM AND CURRENT VALUE WITHOUT QUITTI
      DATA SIGR/5./,SIGB/3./,PI/3.141593/,THRC/.99/,
     *     STEPL/2./,STEPB/4./,KON1/1/,KON2/2/,KON3/3/
      DATA SPGRAN/5./,PSDIF/.08/,CDMIN/10./,CDO/100./,
     *     CDIO/25./,CDMAX/150./, NRUNS/10/,JJNO/8063/
      DATA NDET/4H(ND)/,LCON/4H(LC)/,MCON/4H(MC)/,
     *   ICON/4H(HC)/
      DATA T1 /.20/, T2 /.75/
      DATA P1/6*-1.0/,P2/6*-1.0/,P3/6*-1.0/,HT1/6*-1.0/,
     *     HT2/6*-1.0/,HT3/6*-1.0/,TD/6*-1.0/,CCD/1.0/
      DATA MYSEED /1333567/, KKK /99947/, JCOUNT /0/,
     *NBSEED/356019/,NBMAX/0/
      END
```

CASW COMPUTER SIMULATION PROGRAM FOR TASDA

```
C
C
C          MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      COMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C          NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C          MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C          NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C          MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C          (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C          MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/ IDENT(4,40)
C          MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JPLOT(41)
C          AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/ M,ISTEP,      J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
      COMMON/B/ JCOUNT,KKK,MYSEED,NBSEED,JJJ,NBMAX
      COMMON/C/ NSTEP
C
C
C
      RN = 0.0
      CALL INDAT
998   FORMAT (8F10.3)
999   FORMAT(9I5)
      READ(5,999) LPT,NHOWMV,NPLIND,NPTS,MINB,MAXB,NRUNS
      READ (5,998) TMELTE,RFRNGE
      READ(5,998)VEL
      READ (5,998) XS1,YS1,XS2,YS2,XF1,YF1,XF2,YF2
      READ(5,998)HDEPTH,SDEPTH,FM
      READ(5,998) STEPL,STEPB
      IF (NPLIND .NE. 1) GO TO 10
      IF (NPTS .LE. 205) GO TO 9
      WRITE (6,2032) NPTS
2032  FORMAT (3X, I5)
      GO TO 66
9     READ (5,998) (PLC(I),I=1,NPTS)
```

```
      READ(5,999)NSTEP
10    IF (LPT .EQ. 2) GO TO 20
      READ (5,998) THRESH
      GO TO 30
20    THRESH = THRC
   30 CONTINUE
2033  FORMAT (I5)
      WRITE(6,997)LPT,NHOWMV,NPLIND,NPTS,MINB,MAXB,
     *             TMELTE,RFRNGE,VEL,XS1,YS1,XS2,YS2,
     *             XF1,YF1,XF2,YF2,THRESH,FM,NRUNS
 997  FORMAT(1X,5HLTP =,I5,3X,8HNHOWMV =,I5,3X,
     1      8HNPLIND =,I5,3X,6HNPTS =,I5,3X,6HMINB =,I5,3X,
     2      6HMAXB =,I5,/,1X,8HTMELTE =,F10.2,3X,
     3      8HRFRNGE =,F10.2,/,1X,5HVEL =,F10.2,2X,/,
     4      2X,20HSTARTING COORDINATES,/,4F20.2,/,
     5     1X,18HENDING COORDINATES,/,4F20.2/,1X,8HTHRESH =
     6     ,F10.2,3X,/,16H FIGURE OF MERIT,F20.2,
     7     /' NUMBER OF RUNS PER PROBABILITY CALC =',I5)
      WRITE(6,996)NSTEP,STEPL,STEPB
996   FORMAT(' NSTEP = ',I5,' STEPL = ',F7.2,' STEPB = ',F7.
C
C        END OF DATA INITIALIZATION
C
      ANPTS=NPTS
      ANRUNS=NRUNS
      VELM=VEL/60.0
      VELAVG = VEL
C        DF = APPROXIMATE DISTANCE MOVED BY TARGET BEFORE
C        AIRCRAFT ARRIVES ON STATION
      DF = TMELTE*VELAVG*.0166666667
      IF (LPT .EQ. 3) GO TO 101
      GO TO (85, 90, 95), NHOWMV
85    DIRMAX = XF1
      DIRMIN = YF1
      DISMAX = XF2
      CDX = .5*AMAX1(ABS(XS1 - XS2), ABS(YS1 - YS2)) + DF +
      GO TO 100
90    CDX1 = AMAX1(XS1,XS2,XF1,XF2) - AMIN1(XS1,XS2,XF1,XF2)
      CDX2 = AMAX1(YS1,YS2,YF1,YF2) - AMIN1(YS1,YS2,YF1,YF2)
      DISMAX = SQRT(CDX1*CDX1 + CDX2*CDX2) - DF
      CDX = .5*AMAX1(CDX1,CDX2)
      GO TO 100
95    CDX = .5*XS1
      DISMAX = YS1
      XS1 = -CDX
      XS2 = CDX
      YS1 = XS2 + DF
      YS2 = YS1
      XF1 = XS1
      XF2 = XS2
      YF1 = XF1
      YF2 = YF1
      GO TO 100
101   DISMAX = XS1
C
C RUNT=AVERAGE TRANSIT TIME WHEN TARGET MOVES DISMAX
100   RUNT = DISMAX/VELAVG
      STEP = 120.0/VEL
      ISTEP = STEP
      NSEGMX = (RUNT*60.)/STEP + 10.
27    IF (NSEGMX .LE. NSGMAX) GO TO 271
C     ABORT ALERT
      GO TO 66
271   CONTINUE
170   KN = 0
      ICFLG = 0
      GO TO (180, 180, 9000), LPT
180   CALL TROP
      GO TO 66
9000  CALL AMAX
66    WRITE (6,9997)
9997  FORMAT (11H END OF RUN)
```

```
      WRITE(6,8866) JCOUNT
8866  FORMAT('  THE NUMBER OF TIMES THE RANDOM NUM. GEN. CAL
      STOP
      END





      SUBROUTINE TROP
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      COMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C        MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JFLCT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/ M,ISTEP,       J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,ICUT,FMCT,NNFD,JJN
C
C
C        INITIALIZATION FOR THRESHOLD/OPTIMUM PROBLEM
C
      STEPL = -1.0
184   IGFLG =1
      CALL GEOM
185   IF (GEOEND) RETURN
      IF (NHOWMV .GE. 2) GO TO 187
      SP = SPGRH(IGE)*CDX
      SPI = SPIGRH(IGE) * CDX
      GO TO 188
187   SP = SPGRT(IGE) * CDX
      SPI = SPIGRT(IGE) *CDX
188   DO 210 K = 1,4
210   JPM(K) = K
      CALL TRSH
      IF (IOFLG .LT. 0) RETURN
```

```
C          *********************
C          *********************
 13    K=2
       IF (LS.EQ.8) K=4
       K = JPM(K)
       ICUT = 0
       ICFLG = 3
       CALL OPUT
       WRITE (6,7) IOFLG
7      FORMAT (4H G  ,5X,7HICFLG =, I5)
C
       ICFLG = 1
       CALL OPUT
310    GO TO 184
       END


       SUBROUTINE PROB
C
C          MAX. NO. OF BUOYS IS 16
       COMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
      *          SE(16)
       COMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
      *FM,TDM,HTM1,HTM2,HTM3,SP,
      *SPI,SPF,SPO,SP1,CD,CDI,
      *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C          NUMBER OF POSITION INDEXES IS 6
       COMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
      *HT3(6),TD(6)
C          MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
       COMMON/A/ PLC(205)
       COMMON/A/ JZONET(12)
C          NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
       COMMON/A/ IDT(12)
       COMMON/A/ KGC(17,2,2)
C          MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C          (INCLUDING REPETITIONS) IS 99
       COMMON/A/ KGI(99)
C          MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
       COMMON/A/ IDENT(4,40)
C          MAXIMUM NO. OF GEOMETRIES IS 40
       COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
      *          SPIGRH(40),JBUOY(41),JPLOT(41)
C          AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
       COMMON/A/ KXB(480),KYB(480)
       COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
      *          STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
      *          CDMAX,CDO,CDIO,
      *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
       COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
      *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
      *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
      *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
      *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
       COMMON/A/ M,ISTEP,      J,K
       COMMON/A/ IOFLG,IGFLG
       LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
       COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
       COMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
      *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
       COMMON/B/ JCOUNT,KKK,MYSEED,NBSEED,JJJ,NBMAX
       COMMON/C/ NSTEP
       DIMENSION DLARGE(16),DSMALL(16)
C
C
C
       NFD = 0
       TDM = 0.
       NHT1 = 0
       NHT2 = 0
       NHT3 = 0
       HTM1 = 0.
```

```
      HTM2 = 0.
      HTM3 = C.
      AK1 = 0.
      AK2 = 0.
      AK3 = 0.
      DO 140 I = 1,NRUNS
      HOLD1 = .FALSE.
      HOLD2 = .FALSE.
      HOLD3 = .FALSE.
      JJ1 = 0
      JJ2 = 0
      JJ3 = 0
      JJN = JJN * KKK
      RN = 0.5 + FLOAT(JJN) * 2.328306E-10
      JCOUNT = JCOUNT + 1
      XS=XS1+RN*(XS2-XS1)
      JJN = JJN * KKK
      RN = 0.5 + FLOAT(JJN) * 2.328306E-10
      JCOUNT = JCOUNT + 1
      YS=YS1+RN*(YS2-YS1)
      IF(NHOWMV.EQ.1)GO TO 16
      JJN = JJN * KKK
      RN = 0.5 + FLOAT(JJN) * 2.328306E-10
      JCOUNT = JCOUNT + 1
      XF=XF1+RN*(XF2-XF1)
      JJN = JJN * KKK
      RN = 0.5 + FLOAT(JJN) * 2.328306E-10
      JCOUNT = JCOUNT + 1
      YF=YF1+RN*(YF2-YF1)
CCALCULATE TARGET BEARING WITH X AXIS-THPX(THROUGH STEP 18)
      IF((XF.GT.XS).AND.(YF.EQ.YS))GO TO 11
      IF((XF.LT.XS).AND.(YF.EQ.YS))GO TO 12
      IF((XF.EQ.XS).AND.(YF.GE.YS))GO TO 1
      THP=ATAN (ABS((XF-XS)/(YF-YS)))
      IF((XF.GT.XS).AND.(YF.GT.YS))GO TO 2
      IF((XF.GE.XS).AND.(YF.LT.YS))GO TO 3
      IF((XF.LT.XS).AND.(YF.LT.YS))GO TO 4
      IF((XF.LT.XS).AND.(YF.GT.YS))GO TO 5
    1 TH=0.0
      GO TO 7
    2 TH=+THP
      GO TO 7
    3 TH=PI-THP
      GO TO 7
    4 TH=PI+THP
      GO TO 7
    5 TH=2.0*PI-THP
      GO TO 7
   11 TH=PI/2.0
      GO TO 7
   12 TH=3.0*PI/2.0
    7 IF(TH.LE.6.29)GO TO 8
      ICFLG = -1
      RETURN
    8 THY=TH*57.2957795
      THPX=(PI/2.0)-TH
      GO TO 28
16    JJN = JJN * KKK
      RN = 0.5 + FLOAT(JJN) * 2.328306E-10
      JCOUNT = JCOUNT + 1
      THY=DIRMIN+RN*(DIRMAX-DIRMIN)
      THPX=PI/2.0-THY/57.2957795
28    CONTINUE
      BC = COS(THPX)
      BS = SIN(THPX)
71    DELTAD = VELM * STEP
      DELTAX = BC*DELTAD
      DELTAY = BS*DELTAD
      SUM = 0.0
      DO 96 J=1,12
      JJN = JJN*KKK
      RN = 0.5+FLOAT(JJN)*2.328306E-10
```

```
      SUM = SUM+RN
96    CONTINUE
      JCCUNT = JCOUNT +12
      FMDR = SIGR*(SUM-6.0)
      DMAX = C.0
      DMIN = DISMAX
      K = NBMAX - NB
      IF(K.EQ.0) GOTO 100
      K = K * 12
      DC 18 L = 1,K
      JJJ = JJJ * KKK
18     CONTINUE
100   DC 99 K = 1,NB
      SUM = 0.0
      DC 40 L=1,12
      JJN = JJN * KKK
      RN = 0.5+FLOAT(JJN)*2.328306E-1C
      SUM = SUM +RN
40    CCNTINUE
      JCCUNT = JCOUNT + 12
      FMDB(K) = SIGB*(SUM - 6.0)
      FMF(K) = FM + FMDR + FMDB(K)
      XDIS = XS - XB(K)
      YDIS = YS - YB(K)
      DISTQ = (BS*XDIS - BC*YDIS)**2
      IF (NPLIND.EQ.0) GOTO 30
      R = ANPTS-1.0
      DISC = R**2 - DISTQ
      IF (DISC.LT.0.0) GOTO 95
30    ANFM = FMF(K) + 2.0
      IF (NPLIND.EQ.0) GOTO 64
      NSUB = NSTEP
      LSUB = NSUB - 1
      IF(NSTEP.LE.1) LSUB = -10
      NR = NPTS - NSUB
62    IF(ANFM.GE.PLC(NR)) GOTC 164
      NR = NR - NSUB
      IF(NR.GT.0) GOTO 62
      IF (NSUB.EQ.1) GOTC 95
163   NSUB = 1
      NR = NR + LSUB
      GCTO 62
164   IF(NSUB.EQ.LSUB + 1) GOTO 163
63    R = NR - 1
      R = R + ((ANFM-PLC(NR))/(PLC(NR+1) -PLC(NR)))
      GOTO 66
64    R = 1
      IF (ANFM.GT.66.0) R = 10**((ANFM-66.0)/17.0)
66    DISC = R**2 - DISTQ
      IF(DISC.LT.0) GOTO 95
      DISC = SQRT(DISC)
      XDIS = XDIS*BC + BS*YDIS
      DLARGE(K) = DISC - XDIS + DELTAD
      DSMALL(K) = -DISC - XDIS - DELTAD
      IF(DSMALL(K).GE.0.0) GOTO 67
      DSMALL(K) = 0.0
      IF(DLARGE(K).LT.0.0) GOTO 95
67    IF (DLARGE(K).GT. DISMAX) DLARGE(K) = DISMAX
      IF (DMAX.LT.DLARGE(K))DMAX = DLARGE(K)
      IF (DMIN.GT.DSMALL(K)) DMIN = DSMALL(K)
      GCTC 99
C    SET DLARGE(K) AS FLAG TC NOT LOOK AT BUOY DURING SEG SEA
95    CLARGE(K) = -1.0
99    CCNTINUE
      IF(DMIN.LT.DF) DMIN = DF
      DIS = DMAX - DMIN
      IF(DIS.LE.C.0) GOTC 140
      IF (DIS.GT.DISMAX) DIS = DISMAX
C    FINC X AND Y CCORDINATES OF SUB STARTING POSIT
      XS = XS + BC* DMIN
      YS = YS + BS * DMIN
      TSTART = (DMIN-DF)/VELM
```

```
      NSEG = (DIS/VELM)/STEP + 1.0
 190  INDALT = 0
      DO 50 J=1,NSEG
  70  II = 0
31    XS = XS + DELTAX
      YS = YS + DELTAY
      DMIN = DMIN + DELTAD
      JFLAG = -1
61    DO 60 N = 1,NB
      IF (DMIN.GT.DLARGE(N)) DLARGE(N) =-1.0
      IF (DLARGE(N).LT.0.0) GOTO 60
C
C   FLAG SET TO INDICATE THAT THERE ARE MORE BUOYS TO BE TES
C
      JFLAG = 1
      IF (DMIN.LT.DSMALL(N)) GOTO 60
77    C(N) = SQRT((XB(N) - XS)**2 + (YB(N)-YS)**2)
      IF (NPLIND.EQ.0) GOTO 43
      IF (C(N).GE.ANPTS - 1.0) GOTO 60
      IC = C(N)
      AIC = IC
      PL(N) = PLC(IC+1) +(PLC(IC+2)-PLC(IC+1))*(C(N)-AIC)
      GOTO 46
43    IF (C(N).GT.1.0) GOTO 44
      PL(N) = 66.0
      GOTO 46
44    PL(N) = 17.0*ALOG10(C(N))+66.0
C
C   COMPUTE SIGNAL EXCESS FOR EACH BUOY
 46   SE(N) = FMF(N) - PL(N)
C
C   INCREASE FOM BY 2 DB FOR ALERTED OPERATOR
C
      IF(INDALT.GE.J)SE(N) = SE(N) + 2.0
 120  IF(SE(N).GT.0.0)II = II + 1
60    CONTINUE
      IF(JFLAG.LT.0) GOTO 69
C
      IA=II+1
      IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 240
      GO TO (128, 124, 122), IA
C         II .GE. 3
 240  HOLD3 = .TRUE.
      HTM3 = HTM3 + STEP
      JJ3 = 1
C         II .GE. 2
 122  HOLD2 = .TRUE.
      HTM2 = HTM2 + STEP
      JJ2 =1
C         II .GE. 1
 124  HOLD1 = .TRUE.
      HTM1  = HTM1 + STEP
      INDALT = J + 0.5 + 20./STEP
      IF(JJ1.EQ.1) GO TO 126
      TDM = TDM + TSTART + J*STEP
      NFD = NFD + 1
      JJ1 = 1
126   IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 134
      GO TO (130,130,132), IA
 128  IF(.NOT. HOLD1) GOTO 130
      NHT1 = NHT1 +1
      HOLD1 = .FALSE.
 130  IF(.NOT.HOLD2) GO TO 132
      NHT2 = NHT2 + 1
      HOLD2 = .FALSE.
 132  IF(.NOT.HOLD3) GO TO 134
      NHT3 = NHT3 + 1
      HOLD3 = .FALSE.
134   CONTINUE
C
   50 CONTINUE
 69   IF (.NOT. HOLD1) GO TO 51
```

```
      NHT1 = NHT1 + 1
      IF(.NOT. HOLD2) GOTO 51
      NHT2 = NHT2 + 1
      IF(.NOT.HOLD3) GOTO 51
      NHT3 = NHT3 + 1
51    CONTINUE
      IF(JJ3.EQ.1) GOTO 55
      IF(JJ2.EQ.1) GOTO 65
      IF(JJ1.EQ.1) GOTO 75
      GO TO 137
 55   AK3 = AK3 + 1.0/ANRUNS
 65   AK2 = AK2 + 1.0/ANRUNS
 75   AK1 = AK1 + 1.0/ANRUNS
137   CONTINUE
  140 CONTINUE
      IF(NFD.EQ.0) GOTO 145
      TDM = TDM/FLOAT(NFD)
 145  IF(NHT1.EQ.0) GOTO 155
      HTM1 = HTM1/FLOAT(NHT1)
      IF(NHT2.EQ.0) GOTO 155
      HTM2 = HTM2 / FLOAT (NHT2)
      IF(NHT3.EQ.0) GOTO 155
      HTM3 = HTM3/FLOAT(NHT3)
155   CONTINUE
      RETURN


      SUBROUTINE TRSH
C
C         MAX. NO. OF BUOYS IS 16
      COMMON/A/  XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *           SE(16)
      COMMON/A/  II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C         NUMBER OF POSITION INDEXES IS 6
      COMMON/A/  JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C         MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/  PLC(205)
      COMMON/A/  JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/  IDT(12)
      COMMON/A/  KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/  KGI(99)
C         MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/  IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/  RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *           SPIGRH(40),JBUOY(41),JPLOT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/  KXB(480),KYB(480)
      COMMON/A/  NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *           STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
     *           CDMAX,CDO,CDIO,
     *           NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/  L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *           YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *           ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *           NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *           IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/  M,ISTEP,         J,K
      COMMON/A/  IOFLG,IGFLG
      LOGICAL    GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/  GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/  BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *           BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
      COMMON/B/  JCOUNT,KKK,MYSEED,NBSEED,JJJ,NBMAX
C
```

```
C
C
C
C
C        SUBROUTINE THRESHOLD/OPTIMIZATION
 1000 LIM = .FALSE.
 1100 LS = 0
1080  LIM1 = .FALSE.
      IF (LPT .LT. 3) GO TO 1010
      IF(LAS.GE.5) GOTO 1210
C
C COMPUTE COORDINATES OF STARTING AND ENDING RECTANGLES
      IF (NHOWMV .EQ. 1) GO TO 1090
      XS2 = CDD
      XS1 = -XS2
      XF2 = XS2
      XF1 = XS1
      YS1 = XS2 + DF
      YS2 = YS1
      YF1 = XS1
      YF2 = XS1
      GO TO 1010
C
 1090 XS2 = CDD - DF - DISMAX
      XS1 = -XS2
      YS1 = XS1
      YS2 = XS2
C
 1010 LSM = LS
      IF (LSM .GE. 8) GO TO 1210
      SPIM = SPI
      IF((LSM.NE.3).AND. .NOT. LIM) GOTO 1015
      IF (ABS(SPIM) .GT. SPMIN) GO TO 1015
      LS = 9
      GO TO 1210
C
 1015 XSP = SP
      IF (LSM .NE. 0) XSP = XSP + SPIM
      IF (LSM - 3) 1031,1018,1040
1018  LT = 3
      WRITE (6,2) LT
2     FORMAT (4H P   ,5X,4HLT =, I5)
      CALL REFN
1020  IF (L.EQ.0) GOTO 1040
      LS = 9
      GO TO 1210
C
C
1031  IF (XSP .LT. SPMIN) GO TO 1032
      IF (XSP .GT. SPMAX) GO TO 1033
      GO TO 1040
C
1032  SPLIM = SPMIN
      GO TO 1034
1033  SPLIM = SPMAX
1034  IF ((LSM.LE.0).OR.(LSM.GE.3)) GO TO 241
      GO TO (1035,1037),LSM
C
C         LSM = 0
  241 SP = SPLIM
      XSP = SPLIM
      GO TO 1040
C
C         LSM = 1
1035  IF (LIM1) GO TO 1036
      SPI = -SPIM
    . LIM1 = .TRUE.
      GO TO 1010
 1036 SPI = .5*SPIM
      GO TO 1010
C
C         LSM = 2
```

```
 1037 SPDIF = SPLIM - SP
      SPIM = .5*SPIM
      LIM = .TRUE.
      IF (ABS(SPDIF) .GE. ABS(SPIM)) GC TO 1038
      SPI = -SPIM
      XSP = SP - SPIM
      LSM = 5
      GC TO 1040
1038  SPI = SPDIF
      XSP = SPLIM
      LSM = 6
C
1040  KK = JPM(4)
C CALCULATE BUCY POSITIONS
      IND = IB1
      DC 1070 J=1,NB
      XB(J) = FLOAT(KXB(IND))*XSP
      YB(J) = FLOAT(KYB(IND))*XSP
1070  INC = IND + 1
      BSPF = BUOYSP *XSP
C
C PERFORM PROBABILITY CALCULATIONS
      JJJ = MYSEED
      JJN = NBSEED
100   CALL PROB
      IF (IOFLG .LT. 0) RETURN
C SAVE VALUES
 1110 P1(KK) = AK1
      P2(KK) = AK2
      P3(KK) = AK3
      HT1(KK) = HTM1
      HT2(KK) = HTM2
      HT3(KK) = HTM3
 1120 TC(KK) = TDM
C
C DC THRESHOLD CALCULATION
130   LT = 1
      CALL REFN
1130  IF (L.EQ.0) GOTO 1170
C  L=1 IMPLIES THAT PROBABILITIES EXCEED THRESHCLD
      LS = 8
      SP = XSP
      GCTO 1210
C
1170  IF (LSM .EQ. 0) GO TO 1190
C DC CCMPARISCN TEST
      LT = 2
      CALL REFN
C UPDATE STATE OF SET M1
 118C IR = 2*LSM - L
      ASSIGN 1185 TO IUPD
      GC TO 3000
1185  GOTO 1200
C
C UPCATE FOR STATE 0 -- COLD START
 119C JPM(4) = JPM(2)
      JPM(2) = KK
      LS = 1
1200  CCNTINUE
      GC TO 1010
 1210 RETURN
C
C
C     SUBROUTINE UPDATE
C
 3000 K = JPM(4)
      IF((IR.LT.1).OR.(IR.GT.14)) GO TO 3020
      GCTO (3030,3020,3030,3040,3050,3060,3070,3C80,3070,
     *       3020,3083,3020,3070,3087),IR
C
 3020 JPM(4) = JPM(1)
      JPM(1) = K
```

```
      SPI = -SPI
      LS = 2
      GC TO 3090
 3030 JPM(4) = JPM(1)
      JPM(1) = JPM(2)
      JPM(2) = K
      SP = SP + SPI
      LS = 2
      GC TO 3090
 304C JPM(4) = JPM(3)
      JPM(3) = K
      SPI = .5*SPI
      LS = 3
      GC TO 3090
 3050 JPM(4) = JPM(1)
      JPM(1) = JPM(2)
      JPM(2) = K
      SP = SP + SPI
      SPI = .5*SPI
C NC CHANGE IN STATE
      GC TO 3090
 306C JPM(4) = JPM(3)
      JPM(3) = K
      SPI = -SPI
      LS = 4
      GC TO 3090
 307C JPM(4) = JPM(3)
      JPM(3) = JPM(2)
      JPM(2) = K
      SP = SP + SPI
      SPI = -.5*SPI
      LS = 3
      GC TO 3C90
 3080 JPM(4) = JPM(1)
      JPM(1) = K
      SPI = -.5*SPI
      LS = 3
      GC TO 3090
 3083 JPM(4) = JPM(1)
      JPM(1) = JPM(2)
      JPM(2) = K
      SP = SP + SPI
      SPI = -.5*SPI
      LS = 7
      GO TO 3090
 3087 JPM(4) = JPM(1)
      JPM(1) = K
      SPI = .5*SPI
C NC CHANGE IN STATE
3090  GO TO IUPD, (1185, 1200)
      END




      SUBROUTINE REFN
C     SUBROUTINE REFINE
C
C
C        MAX. NO. OF BUOYS IS 16
      CCMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *           SE(16)
      CCMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CCD,IGEA,NFO,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C        NUMBER OF POSITION INDEXES IS 6
      CCMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
```

```
C         MAX. NO. OF POINTS IN PROPOGATICN LOSS CURVE IS 205
      CCMMON/A/ PLC(205)
      CCMMON/A/ JZCNET(12)
C         NO. OF CHARACTERS IN TACTICS ICENTIFICATION IS 60
      CCMMON/A/ IDT(12)
      CCMMON/A/ KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATICN LIST
C         (INCLUDING REPETITIONS) IS 99
      CCMMON/A/ KGI(99)
C         MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      CCMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GECMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JPLCT(41)
C         AVERAGE NO. OF BUCYS IN A GECMETRY IS 12
      CCMMON/A/ KXB(480),KYB(480)
      CCMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSCIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCCN,ICON,T1,T2
      CCMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      CCMMON/A/ M,ISTEP,      J,K
      CCMMON/A/ IOFLG,IGFLG
      LCGICAL   GEOEND,AC,HOLD1,HOLD2,HCLD3,LIM,LIM1,LFA
      CCMMON/A/ GEOEND,AC,HOLD1,HOLD2,HCLD3,LIM,LIM1,LFA
      CCMMON/A/ BSPF,BTCM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
C
C
 5000 L = 0
      IF((LT .LT. 0).OR.(LT .GT. 3))GO TO 5020
      GC TO (5010, 5020, 5030), LT
C
C     LT = 1, THRESHOLD TEST
 5010 IF (AK1.GT.THRESH) L = 1
      RETURN
C
C         LT = 2 OR LT .GT. 3, COMPARISCN TESTS
C         KK HAS INITIALLY BEEN SET TO JPM(4),(FCR
C         THRESHOLD/OPTIMUM) OR TO  JPM(L),L = 5 CR 6
C         (FCR AREA MAXIMIZATION)
 5020 K = JPM(LT)
      IF (P1(KK).GT.P1(K)) L = 1
      RETURN
C
C         LT = 3, OPTIMUM TEST
 5030 K = JPM(1)
      D = P1(K)
      K = JPM(2)
      E = P1(K)
      K = JPM(3)
      F = P1(K)
      IF (((D-F)**2).LE.(PSDIF*(2.*E-D-F))) L = 1
 5070 RETURN
      END




      SUBROUTINE GEOM
C
C         MAX. NO. OF BUOYS IS 16
      CCMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      CCMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
```

```
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CC1
C         NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C         MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JPLCT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSCIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/ M,ISTEP,        J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
C
C
C
C         GET GEOMETRY DATA SUBROUTINE,IF GEOEND = TRUE
C         NO ADDITIONAL GEOMETRIES ARE AVAILABLE
      IF (IGFLG .EQ. 2) GO TO 6000
4000  IF (KN .NE. 0) GO TO 4110
4100  GEOEND = MINB .GT. MAXB
      IF (GEOEND) GO TO 4060
      N = MINO(NHOWMV,2)
      I1 = KGC(MAXB, 2 , N)
      KN = KGC(MAXB + 1, 2, N) -I1
      MAXB = MAXB - 1
      IF (KN .EQ. 0) GO TO 4100
4110  IGE = KGI(I1)
      I1 = I1 + 1
      KN = KN - 1
C
C     COLLECT GEOMETRY DATA
C
6000  IB1 = JBUOY(IGE)
      NB = JBUOY(IGE + 1) - IB1
      BUOYSP = (KXB(IB1)-KXB(IB1+1))**2 + (KYB(IB1)
     *- KYB(IB1+1))**2
      BUOYSP = SQRT(BUOYSP)
      SPMAX = RFRNGE / RADIUS(IGE)
      SPMIN = SPGRAN / BUOYSP
4060  RETURN
      END
```

```
      SUBROUTINE OPUT
C
C         MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      COMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C         NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C         MAX. NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATION LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
      COMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JPLCT(41)
C         AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSCIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/ M,ISTEP,      J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
C
C
C
C     SUBROUTINE TO PRINT 1ST PART
2004  FORMAT (12H1GEOMETRY ID, 4A4)
2005  FORMAT (30X,21HSPACING (NAUT. MI.) =,1X,F5.0)
2006  FORMAT (20X, 31HMEAN TIME TO DETECTION (MIN.) =,F5.0,
     *1X,A4)
2007  FORMAT (21H MEAN HOLDING TIME OF,I2,
     *28H OR MORE DETECTIONS (MIN.) =,1X,F5.0)
2013  FORMAT (3X, I5, 12I5)
2016  FORMAT(30X,21HFIGURE OF MERIT(DB) =,1X,F7.2)
2018  FORMAT (17H NUMBER OF BUOYS=,I5/)
2021  FORMAT(1X,32HEFFECTIVE RADIUS OF AREA COVERED,1X,
     *14H(NAUT. MI.) = ,F11.2)
2026  FORMAT (//,34H ABBREVIATIONS--ND = NO DETECTION,,1X,
     *44HLC = LOW CONFIDENCE, MC = MEDIUM CONFIDENCE,,1X,
     *20HHC = HIGH CONFIDENCE,///)
2027  FORMAT(1X,14HPROBABILITY OF,I2,1X, 7HOR MORE,1X,
     *24HSIMUTANIOUS DETECTIONS =,1X,F7.2)
2036  FORMAT (22H CONTINUOUS MONITORING)
      IF (IOFLG .EQ. 3) GO TO 8000
7000  WRITE (6,2004) (IDENT(I,IGE),I=1,4)
      WRITE (6,2018) NB
      WRITE (6,2036)
```

```
C
C
C       SUBROUTINE TO PRINT 2ND PART
        IF (LPT .EQ. 3) GO TO 7250
        WRITE(6,2016)FM
        GO TO 7500
 7250   WRITE(6,2016)FMOT
        WRITE(6,2021)CDD
 7500   WRITE(6,2005)BSPF
        WRITE(6,2027) KON1, BAK1
        WRITE(6,2027) KON2, BAK2
        WRITE(6,2027) KON3, BAK3
        WRITE(6,2006) BTDM, NNFD
        WRITE(6,2007) KON1, BHTM1
        WRITE(6,2007) KON2, BHTM2
        WRITE(6,2007) KON3, BHTM3
        WRITE(6,2026)
        RETURN
C
C
C
C       SUBROUTINE TO PREPARE DATA FOR PRINTING
8000    IF (IOUT .EQ. 0) GO TO 8005
 8050   JSAVE = J
        KSAVE = K
        LSAVE = L
        NSAVE = N
 8070   IND = IB1
        DO 8080 I = 1,NB
        XB(I) = FLOAT(KXB(IND))*SP
        YB(I) = FLOAT(KYB(IND))*SP
8080    IND = IND + 1
        IF (LPT .LT. 3) GO TO 8085
        IF (NHOWMV .EQ. 1) GO TO 8083
        XS2 = CDD
        XS1 = -XS2
        XF2 = XS2
        XF1 = XS1
        YS1 = XS2 + DF
        YS2 = YS1
        YF1 = XS1
        YF2 = XS1
        GO TO 8085
 8083   XS2 = CDD - DF - DISMAX
        XS1 = -XS2
        YS1 = XS1
        YS2 = XS2
8085    CALL PRCB
        J = JSAVE
        K = KSAVE
        L = LSAVE
        N = NSAVE
        P1(K) = AK1
        P2(K) = AK2
        P3(K) = AK3
        HT1(K) = HTM1
        HT2(K) = HTM2
        HT3(K) = HTM3
 8090   TD(K) = TDM
        ICUT = 0
 8005   BSPF = BUOYSP * SP
        BTDM = TD(K)
        BHTM1 = HT1(K)
        BHTM2 = HT2(K)
        BHTM3 = HT3(K)
        BAK1  = P1(K)
        BAK2 = P2(K)
        BAK3 = P3(K)
        IF(BTDM.GT.0.) GOTO 8010
        NNFD = NDET
        BTDM = 9999.
        GO TO 8040
```

```
 8010 IF(BAK1.GT.T1) GOTO 8020
      NNFD = LCON
      GC TO 8040
 802C IF(BAK1.GT.T2) GOTO 8030
      NNFD = MCON
      GC TO 8C40
 8030 NNFD =  ICON
8040  RETURN
      END




      SUBROUTINE AMAX
C
C         MAX. NO. OF BUOYS IS 16
      CCMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      CCMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CDD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C         NUMBER OF POSITICN INDEXES IS 6
      CCMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C         MAX. NO. OF PCINTS IN PROPOGATION LOSS CURVE IS 205
      CCMMON/A/ PLC(205)
      CCMMON/A/ JZONET(12)
C         NO. OF CHARACTERS IN TACTICS ICENTIFICATION IS 60
      CCMMON/A/ IDT(12)
      CCMMON/A/ KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATICN LIST
C         (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C         MAX. NO  OF CHARACTERS IN GECM. IDENTIFICATION = 16
      CCMMON/A/ IDENT(4,40)
C         MAXIMUM NO. OF GECMETRIES IS 4C
      CCMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(4C),JBUOY(41),JPLCT(41)
C         AVERAGE NO. OF BUCYS IN A GECMETRY IS 12
      CCMMON/A/ KXB(480),KYB(480)
      CCMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KCN1,KON2,KCN3,SPGRAN,PSCIF,CDMIN,
     *          CDMAX,CDO,CDIC,
     *          NRUNS,JJNO,NCET,LCON,MCCN,ICON,T1,T2
      CCMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,CIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      CCMMON/A/ M,ISTEP,       J,K
      CCMMON/A/ IOFLG,IGFLG
      LCGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      CCMMON/A/ GEOEND,AC,HCLD1,HCLD2,HCLD3,LIM,LIM1,LFA
      CCMMON/A/ BSPF,BTDM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
C
C
C
C         AREA MAXIMIZATICN PROBLEM
 900C CD = CDO
      LAS = 0
      LFA = .FALSE.
      IGEA = 0
      DC 9007 K=1,6
9007  JPM(K) = K
      IF (NHOWMV .GT. 1) GO TO 9010
      CDMIN = AMIN1(CDMIN, DF + DISMAX)
      DIRMIN = 0.
```

```
      DIRMAX = 360.
C
9010  IGFLG = 1
      CALL GECM
9015  IF (GEOEND) GO TO 9320
      IF (NHOWMV .GE. 2) GO TO 9017
      SPR = SPGRH(IGE)
      SPIR = SPIGRH(IGE)
      GO TO 9018
9017  SPR = SPGRT(IGE)
      SPIR = SPIGRT(IGE)
9018  IF(LAS.EQ.8) GOTO 9030
      LAS = 0
      SP = SPR*CD
      SPI = SPIR*CD
      CDD = CD
      CDI = CDI0
9030  CALL TRSH
      IF (IOFLG .LT. 0) RETURN
C
 9045 LSA = 2
      LASM = LAS
      LSM = LS
      KK = JPM(5)
      LT = 2
      J = 2*LASM - LSM + 10
      IF((J.LT.1) .OR. (J.GT.10)) GO TO 9046
      GO TO (9050,9070,9080,9130,9140,9150,9160,9180,9200,92
 9046 LSA = 0
      GO TO 9220
 9050 IF(.NOT. LFA) GOTO 9060
      LAS = 3
      GO TO 9220
 9060 CDI = -CDI
      LAS = 1
      GO TO 9220
 9070 LAS = 4
      GO TO 9215
9080  CALL REFN
C        IF L=1, THE OLD VALUE IS BIGGER THAN THE NEW VALUE
9090  IF (L .EQ. 1) GO TO 9100
      GOTO 9110
 9100 CDI = -CDI
      CD = CD + CDI
      LAS = 3
      LSA = 0
9110  IF (L .EQ. 0) GO TO 9120
      GOTO 9220
 9120 LAS = 3
      LSA = 2
      IF(LFA) LAS = 5
      GO TO 9220
 9130 LAS = 2
      CDI = -CDI
      SP1 = SP
      GO TO 9215
 9140 SP0 = SP
      SPI = 4.*SPI
      LSA = 0
      GO TO 9220
 9150 CD = CDD
      SP1 = SP
      GO TO 9215
9160  CALL REFN
C        IF L=1, THE OLD VALUE IS BIGGER THAN THE NEW VALUE
9170  IF (L .EQ. 1) GO TO 9173
      IF(LFA) LAS = 5
      GOTO 9220
 9173 LAS = 5
      LSA = 0
      CD = CD - CDI
      GOTO 9220
```

```
 9180 IF(CDI.GT.0.) GOTO 9190
      CDI = -CDI
      LAS = 2
      GC TO 9215
 919C LAS = 4
      GC TO 9215
 9200 LAS = 2
      CD = CD - CDI
      SPO = SP
      SPI = 4.*SPI
      LSA = 0
      GC TO 9220
 9210 IF(.NOT. LFA) GOTO 9215
      LAS = 8
      IGEA = IGE
      CD1 = CC
C
 9215 LSA = 4
      SP1 = SP
C
9220  AC = .FALSE.
 9225 LASM = LAS
      L = 5
      IF ((LASM.LE.0).OR.(LASM.GE.6)) GC TO 9226
      GC TO (9230,9270,9230,9230,9283), LASM
9226  L=6
      GC TO 9300
C
9230  AC = .TRUE.
      SPI = 4.*SPI
      CDD = CD + CDI
      LFA = .FALSE.
      IF(CDD.GE.CDMAX) GOTO 9240
      IF(CDD.GT.CDMIN) GCTO 9260
      CDLIM = CDMIN
      GC TO 9250
924C  CDLIM = CDMAX
 925C CDI = CDLIM - CD
      CDD = CDLIM
      LFA = .TRUE.
 926C CD = CDD
      IF (LSA.NE.0) SPO = SP
      SP = SP + SPR*CDI
      GC TO 9300
C
 927C IF(CDI.LE.CDMIN) GCTO 9280
      CDI = CDI*.5
      CDD = CC + CDI
      SP = .5*(SPO + SP1)
      AC = .TRUE.
      GC TO 9300
 928C IGEA = IGE
      CD1 = CD
      LFA = .TRUE.
      LAS = 7
      IF(LSA.EQ.0) LSA = 5
      L = 6
      GO TO 9300
C
9283  LAS = 7
      LT = LSA
      IF (LT .EQ. 0) LT = 5
      LSA = LT
      IF(IGEA) 9286, 9295, 9310
 9286 KK = JPM(6)
      CALL REFN
C         IF L = 1,THE VALUE FROM A PREVIOUS GEOM IS BIGGER
9289  IF (L .EQ. 1) GO TO 9292
      GC TO 9295
 9292 LSA = 0
      GCTC 9310
C
```

```
 9295 IGEA = -IGE
      CD1 = CD
      IF (LT .EQ. 2) GO TO 9298
      SP1 = SPO
      GC TO 9299
 9298 SP1 = SP
9299  L = 6
 9300 J = LSA
      IF (J .EQ. 0) GO TO 9310
C         SAVE VALUES
       K = JPM(L)
      JPM(L) = JPM(J)
      JPM(J) = K
9310  IF (AC) GOTO 9030
C
      GO TO 9010
C
C
9320  IGE = IABS(IGEA)
      IF (IGE .EQ. 0) GO TO 9380
 9330 LS = 0
      N = IABS(IGEA)
      IF (N .NE. IGE) GO TO 9350
      IGEA = 0
      SP = SP1
9350  CCNTINUE
C
      IGFLG = 2
      CALL GECM
9355  L = 0
      IOUT = 0
      FMCT = FM
      CDD = CD1
      K = JPM(6)
      ICFLG = 3
      CALL OPUT
      IOFLG = 1
      CALL OPUT
9380  CCNTINUE
      RETURN
      END
```

```
      SUBROUTINE INDAT
C
C   SUBROUTINE TO READ GEOMETRY DATA INTO PROGRAM
C
C
C         MAX. NO. OF BUOYS IS 16
      CCMMON/A/ XB(16),YB(16),FMDB(16),FMF(16),C(16),PL(16),
     *          SE(16)
      CCMMON/A/ II,JJ1,JJ2,JJ3,AK1,AK2,AK3,INDALT,
     *FM,TDM,HTM1,HTM2,HTM3,SP,
     *SPI,SPF,SPO,SP1,CD,CDI,
     *CCD,IGEA,NFD,LS,LAS,NHT1,NHT2,NHT3,NF,LSA,CD1
C         NUMBER OF POSITION INDEXES IS 6
      CCMMON/A/ JPM(6),P1(6),P2(6),P3(6),HT1(6),HT2(6),
     *HT3(6),TD(6)
C         MAX. NO. OF POINTS IN PROPOGATICN LOSS CURVE IS 205
      CCMMON/A/ PLC(205)
      CCMMON/A/ JZCNET(12)
C         NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      CCMMON/A/ IDT(12)
      CCMMON/A/ KGC(17,2,2)
C         MAX. NO. OF GEOM INDEXES IN CLASSIFICATICN LIST
C         (INCLUDING REPETITIONS) IS 99
      CCMMON/A/ KGI(99)
C         MAX. NO  OF CHARACTERS IN GEOM. IDENTIFICATION = 16
```

```
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ RADIUS(40),SPGRT(40),SPIGRT(40),SPGRH(40),
     *          SPIGRH(40),JBUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,
     *          STEPB,KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,
     *          CDMAX,CDO,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,XS1,XS2,YS1,YS2,XF1,XF2,
     *          YF1,YF2,MINB,MAXB,STEP,VELM,ANPTS,ICHAN,
     *          ANRUNS,KK,LT,M1,NB,FM1,DIRMIN,DIRMAX,DISMAX,
     *          NHOWMV,NPLIND,NSEGMX,RFRNGE,THRESH,
     *          IB1,SPMIN,SPMAX,IGE,MN,BUOYSP
      COMMON/A/ M,ISTEP,      J,K
      COMMON/A/ IPFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ BSPF,BTCM,BHTM1,BHTM2,BHTM3,BAK1,BAK2,
     *          BAK3,ISEED,NPTS,IOUT,FMCT,NNFD,JJN
      DIMENSION KG(4), KGS(4,40,16), KXBT(16), KYBT(16),
     *NGE(40,4)
C     NUMBER OF GEOMETRIES IS  10
      DATA NGMAX/10/
998   FORMAT ('1GEOMETRY NO.',I3/' GEOMETRY ID ',4A4/
     *' NUMBER OF BUOYS',I3//' TARGET APPLICATIONS')
997   FORMAT (5X,4A4)
996   FORMAT ('0',7X,'BUOY  DATA'/'0BUOY NO.    X          Y'/
     *(I5,I9,I8))
994   FORMAT (3(I3, ','), I3)
993   FORMAT (7(I3, ','), I3)
992   FORMAT (11(I3, ','), I3)
985   FORMAT ('ORADIUS =',F9.2/)
1000  FORMAT (4A4,4I1,2I5/ 5F8.2)
1001  FORMAT (16I5)
1002  FORMAT(80A1)
      IGE = 1
      DO 10 I = 1,4
      DO 10 NB = 1, 16
10    NGE(NB, I) = 0
      JBM = 1
      JP = 1
C
90    READ(5,1000,END=220)(IDENT(J,IGE),J=1,4),(KG(I),I=1,4)
     *, NB, NP, SPGRT(IGE), SPIGRT(IGE), SPGRH(IGE),
     *  SPIGRH(IGE), RADIUS(IGE)
      WRITE(6,998) IGE, (IDENT(J,IGE), J=1,4), NB
      DO 100  I = 1,4
      IF (KG(I) .NE. 1) GO TO 100
      J = NGE(NB, I) + 1
      KGS(I, J, NB) = IGE
      NGE(NB,I) = J
100   CONTINUE
C
      JBUOY(IGE) = JBM
      JPLOT(IGE) = JP
C
      READ(5, 1001, END=210) (KXBT(I), KYBT(I), I=1,NB)
      WRITE(6,996) (I, KXBT(I), KYBT(I), I=1,NB)
      J = 1
      LIMT= JBM + NB - 1
      DO 105 I=JBM,LIMT
      KXB(I) = KXBT(J)
      KYB(I) = KYBT(J)
105   J = J + 1
      JBM = LIMT+ 1
C
C
C
C
130   IF (KG(2) + KG(4) .NE. 0) GO TO 1410
```

```
      SPGRT(IGE) = 0.0
      SPIGRT(IGE) = 0.0
1410  IF (KG(1) + KG(3) .NE. 0) GO TO 1420
      SPGRH(IGE) = 0.0
      SPIGRH(IGE) = 0.0
1420  WRITE(6,985) RADIUS(IGE)
      JP = LIMT+ 1
C
C
      IGE = IGE + 1
      IF (IGE .LE. NGMAX) GO TO 90
C
      JBUOY(IGE) = JBM
      JFLCT(IGE) = JP
C
      K = 1
      L = 1
      DO 260 I=1,2
      DO 250 J=1,2
      DO 240 NB=1,16
      KGC(NB, I, J) = L
      N = NGE(NB,K)
      IF (N .EQ. 0) GO TO 240
      L1 = L
      DO 230 M=1,N
      KGI(L) = KGS(K, M, NB)
      L = L + 1
230   CONTINUE
C
      LIMT= L - 1
240   CONTINUE
C
      KGC(17, I, J) = L
250   K = K + 1
C
260   CONTINUE
C
      IGE = IGE - 1
C
      JBM = JBM - 1
      JP = JP - 1
      L = L - 1
C
210   CONTINUE
220   RETURN
      END
```

COMPUTER PROGRAM LISTING C

MODIFIED PROB SUBROUTINE

```
      SUBROUTINE PROB
C
C        MAX. NO. OF BUOYS IS 16
      COMMON/A/ XB(16),YB(16),NOW(400,16),FMDB(16),FMF(16),C(16),PL(16),
     1          SE(6,16),MON(16)
C        MAXIMUM NO. OF FIGURES-OF-MERIT IS 6
      COMMON/A/ FMD(6),II(6),JJ1(6),JJ2(6),JJ3(6),AK1(6),AK2(6),AK3(6),
     *          INDALT(6),FM(6),TDM(6),HTM1(6),HTM2(6),HTM3(6),SF(6),
     *          SPI(6),SPF(6),SPO(6),SP1(6),JFMI(6,6),CD(6),CDI(6),
     *          CDD(6),IGEA(6),NFD(6),LS(6),LAS(6),NHT1(6),NHT2(6),
     *          NHT3(6),NF(6),       LSA(6),CD1(6)
C        NUMBER OF POSITION INDEXES IS 6
      COMMON/A/ JPM(6,6),P1(6,6),P2(6,6),P3(6,6),HT1(6,6),HT2(6,6),
     *          HT3(6,6),TD(6,6)
C        MAXIMUM NO. OF POINTS IN PROPOGATION LOSS CURVE IS 205
      COMMON/A/ PLC(205)
      COMMON/A/ JZONET(12)
C        NO. OF CHARACTERS IN TACTICS IDENTIFICATION IS 60
      COMMON/A/ IDT(12)
      COMMON/A/ KGC(17,2,2)
C        MAXIMUM NO. OF GEOMETRY INDEXES IN CLASSIFICATION LIST
C        (INCLUDING REPETITIONS) IS 99
      COMMON/A/ KGI(99)
C        MAXIMUM NO. OF CHARACTERS IN GEOMETRY IDENTIFICATION IS 16
      COMMON/A/ IDENT(4,40)
C        MAXIMUM NO. OF GEOMETRIES IS 40
      COMMON/A/ TMN(40,3),RADIUS(40),SFGRT(40),SFIGRT(40),SPGRH(40),
     1          SPIGRH(40),JZONE(41,3),JBUOY(41),JPLOT(41)
C        AVERAGE NO. OF BUOYS IN A GEOMETRY IS 12
      COMMON/A/ KXB(480),KYB(480)
C        AVERAGE NO. OF ZONES FOR 4 CHANNELS IS LESS THAN 5
      COMMON/A/ IZONE4(4,200)
C        AVERAGE NO. OF ZONES FOR 8 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONE8(8,120)
C        AVERAGE NO. OF ZONES FOR 12 CHANNELS IS LESS THAN 3
      COMMON/A/ IZONEC(12,100)
      COMMON/A/ NPMAX,NSGMAX,ITM,SIGR,SIGB,PI,THRC,STEPL,STEPB,
     *          KON1,KON2,KON3,SPGRAN,PSDIF,CDMIN,CDMAX,CDC,CDIO,
     *          NRUNS,JJNO,NDET,LCON,MCON,ICON,T1,T2
      COMMON/A/ L,DF,KN,CDX,LPT,MFM,NFM,XS1,XS2,YS1,YS2,XF1,XF2,YF1,YF2,
     *          MINB,MAXB,MODE,STEP,VELM,ANPTS,ICHAN,NCHAN,ANRUNS,
```

```
     *              DIRMIN,DIRMAX,DISMAX,NFOWMV,NPLINC,NSEGMX,RFRNGE,THRESH,
     *              TSNMIN,TSNMAX,TSUMIN,TSUMAX,VELSNM,KK,LT,M1,NA,NB,FM1,
     *              IB1,NFMS,SPMIN,SPMAX,IGE,MN,TMON,EUCYSP,JZONEO,JZONE1
      COMMON/A/ M,ISTEP,       J,K
      COMMON/A/ IOFLG,IGFLG
      LOGICAL   GEOEND,AC,HOLD1,HOLD2,HOLD3,LIM,LIM1,LFA
      COMMON/A/ GEOEND,AC,HOLD1(6),HOLD2(6),HOLD3(6),LIM(6),LIM1
      COMMON/A/ LFA(6)
      COMMON /A/BSPF(6), BTDM(6), BHTM1(6), BHTM2(6), BHTM3(6), BAK1(6),
     1          BAK2(6), BAK3(6),ISEED(6), NPTS, ISPEED, ICUT, FMCT(6),
     2          NNFD(6), JJN
      DIMENSION DLARGE(16),DSMALL(16)
C
C
C
81    DO 150 KXY=1,NFM
      NFD(KXY) = 0
      TDM(KXY) = 0.
      NFT1(KXY) = 0
      NFT2(KXY) = 0
      NFT3(KXY) = 0
      HTM1(KXY) = 0.
      HTM2(KXY) = 0.
      HTM3(KXY) = 0.
      AK1(KXY)=0.0
      AK2(KXY)=0.0
150   AK3(KXY)=0.0
      DO 140 I = 1,NRUNS
      DO 160 LXY=1,NFM
      HOLD1(LXY) = .FALSE.
      HOLD2(LXY) = .FALSE.
      HOLD3(LXY) = .FALSE.
      JJ1(LXY)=0
      JJ2(LXY)=0
160   JJ3(LXY)=0
      CALL RADN (RN, JJN)
      XS=XS1+RN*(XS2-XS1)
      CALL RADN (RN, JJN)
      YS=YS1+RN*(YS2-YS1)
      IF(NFOWMV.EQ.1)GO TO 16
      CALL RADN (RN, JJN)
      XF=XF1+RN*(XF2-XF1)
      CALL RADN (RN, JJN)
      YF=YF1+RN*(YF2-YF1)
CCALCULATE TARGET BEARING WITH X AXIS-THPX(THROUGH STEP 18)
      IF((XF.GT.XS).AND.(YF.EQ.YS))GO TO 11
      IF((XF.LT.XS).AND.(YF.EQ.YS))GO TO 12
      IF((XF.EQ.XS).AND.(YF.GE.YS))GO TO 1
```

```
      THP=ATAN (ABS((XF-XS)/(YF-YS)))
      IF((XF.GT.XS).AND.(YF.GT.YS))GO TO 2
      IF((XF.GE.XS).AND.(YF.LT.YS))GO TO 3
      IF((XF.LT.XS).AND.(YF.LT.YS))GO TO 4
      IF((XF.LT.XS).AND.(YF.GT.YS))GO TO 5
    1 TH=0.0
      GO TO 7
    2 TH=+THP
      GO TO 7
    3 TH=PI-THP
      GO TO 7
    4 TH=PI+THP
      GO TO 7
    5 TH=2.0*PI-THP
      GO TO 7
   11 TH=PI/2.0
      GO TO 7
   12 TH=3.0*PI/2.0
      GO TO 7
    7 IF(TH.LE.6.29)GO TO 8
      ICFLG = -1
      RETURN
    8 THY=TH*57.2957795
      THPX=(PI/2.0)-TH
      GO TO 28
16    CALL RADN (RN, JJN)
      THY=DIRMIN+RN*(DIRMAX-DIRMIN)
      THPX=PI/2.0-THY/57.2957795
28    CONTINUE
      BC = COS(THPX)
      BS = SIN(THPX)
C        COMPUTE F.O.M. DEVIATION FOR EACH RUN
      SUM = 0.0
      DO 96 J=1,12
      CALL RADN (RN, JJN)
      SUM = SUM+RN
96    CONTINUE
      FMDR = SIGR*(SUM-6.0)
      DMAX = 0.0
      DMIN = DISMAX
      DO 99 K=1,NB
C        COMPUTE F.O.M. DEVIATION FOR EACH BUOY
      SUM = 0.0
      DO 40 L=1,12
      CALL RADN (RN, JJN)
      SUM = SUM +RN
40    CONTINUE
      FMDB(K) = SIGB*(SUM - 6.0)
```

```
      FMF(K) = FM1 +FMDR + FMCB(K)
C        TEST TO SEE IF SUB WILL COME WITHIN DETECTION RANGE
C        OF BUOY.  IF NOT SET FLAG SO BUOY WILL NOT BE CHECKED
C        DURING SEGMENTED PORTION OF RUN.
      XDIS = XS - XB(K)
      YDIS = YS - YB(K)
      DISTQ = (BS*XDIS - BC*YDIS)**2
      IF (NPLINC.EQ.0) GOTO 30
      R = ANPTS-1.0
      DISC = R**2 - DISTQ
      IF (DISC.LT.0.0) GOTO 95
C   ADD 2.0 FOR POSSIBILITY OF ALERTED OPERATOR
30    ANFM = FMF(K) + 2.0
      IF (NPLINC.EQ.0) GOTO 64
      NR = NPTS
62    IF (ANFM.GE.PLC(NR)) GOTO 63
      NR = NR- 1
      IF (NR.EQ.0) GOTO 95
      GOTO 62
63    R = NR - 1
      R = R + ((ANFM-PLC(NR))/(PLC(NR+1) -PLC(NR)))
      GOTO 66
64    R = 1
      IF (ANFM.GT.66.0) R = 10**((ANFM-66.0)/17.0)
66    DISC = R**2 - DISTQ
      IF(DISC.LT.0) GOTO 95
      DISC = SQRT(DISC)
      XDIS = XDIS*BC + BS*YDIS
      DSMALL(K) = -DISC - XDIS - DELTAD
      DLARGE(K) = DISC - XDIS + DELTAD
      IF(DSMALL(K).GE.0.0) GOTO 67
      DSMALL(K) = 0.0
      IF(DLARGE(K).LT.0.0) GOTO 95
67    IF (DLARGE(K).GT. DISMAX) DLARGE(K) = DISMAX
C        DETERMINE WHERE SUB CAN FIRST BE DETECTED AND
C        WHERE LAST DETECTION LOST.
      IF (DMAX.LT.DLARGE(K))DMAX = DLARGE(K)
      IF (DMIN.GT.DSMALL(K)) DMIN = DSMALL(K)
      GOTO 99
C    SET DLARGE(K) AS FLAG TO NOT LOOK AT BUOY DURING SEG SEARCH
95    DLARGE(K) = -1.0
99    CONTINUE
      IF(DMIN.LT.DF) DMIN = DF
C        DISTANCE EQUALS POINT WHERE DETECTION LOST BY LAST
C        BUOY IN CONTACT MINUS PT. WHERE 1ST BUOY GAINS CONTACT
      DIS = DMAX - DMIN
      IF(DIS.LE.0.0) GOTO 140
      IF (DIS.GT.DISMAX) DIS = DISMAX
```

```
      TSTART = (DMIN-DF)/VELM
      NSEG = (DIS/VELM)/STEP + 1.0
      IF(MODE.EQ.1)GO TO 71                                           503
      CALL RADN (RN, JJN)                                             504
      TSN=TSNMIN+RN*(TSNMAX-TSNMIN)                                   505
      NTSN=TSN/STEP                                                   506
      TSN=NTSN*ISTEP                                                  507
      CALL RADN (RN, JJN)                                             508
      TSU=TSUMIN+RN*(TSUMAX-TSUMIN)                                   509
      NTSU=TSU/STEP                                                   510
      TSU=NTSU*ISTEP                                                  511
      CYC=TSU+TSN                                                     512
      CALL RADN (RN, JJN)                                             513
      BEGCYC=RN*CYC                                                   514
      NSEG=(DIS/(VELM*STEP*TSU/CYC+VELSNM*STEP*TSN/CYC))+1.0          515
CDELTAX,DELTAY=TARGET MOVEMENT PER TIME INCREMENT                     520
C         COMPUTE DELTA DIST. FOR EACH SEGMENT PLUS X AND Y COMPONENTS
71    DELTAD = VELM * STEP
      DELTAX = BC*DELTAD
      DELTAY = BS*DELTAD
      IF(MODE.EQ.1)GO TO 94                                           523
      DELDSN = VELSNM * STEP
      DELXSN = BC * DELDSN
      DELYSN = BS * DELDSN
   94 CONTINUE                                                        526
C    FIND X AND Y COORDINATES OF SUB STARTING POSIT
      XS = XS + BC* DMIN
      YS = YS + BS * DMIN
98    DO 190 IXZ=1,NFM                                                540
  190 INDALT(IXZ)=0                                                   541
C         SET UP PORTION OF TRACK WHERE DETECTION
C         POSSIBLE, THIS WILL CORRESPOND TO ORIGINAL
C         MONITOR CYCLE SET UP IN 'GEOM'
      ISTART = TSTART/STEP + 0.5
      IFINSH = ISTART + NSEG
C         SUB IS STEPPED ALONG THAT PORTION OF TRACK WHERE
C         PROBABILITY OF DETECTION EXISTS
      DO 50 J = ISTART,IFINSH
      DO 70  NXY=1,NFM                                                550
70    II(NXY)=0                                                       551
      AJ = FLOAT(J)
      TREAL = AJ * STEP
      IF(MODE.EQ.1)GO TO 31                                           554
      IA=TREAL/CYC                                                    555
      AIA=IA                                                          556
      TIC=TREAL-AIA*CYC+BEGCYC                                        557
      IF(TIC.GT.CYC)TIC=TIC-CYC                                       558
CTIC=ELAPSED TIME OF PRESENT SUBMERGED/SNORKEL CYCLE                  559
```

```
      IF(TIC.GE.TSU)GO TC 32                                              560
CMCVE TARGET(UPDATE XS,YS)                                                561
      XS=XS+DELTAX                                                        562
      YS=YS+DELTAY                                                        563
      DMIN = DMIN + DELTAD
      GC TO 121                                                           564
   32 XS=XS+DELXSN                                                        565
      YS=YS+DELYSN                                                        566
      DMIN = DMIN + DELDSN
      GC TO 61                                                            567
31    XS = XS + DELTAX
      YS = YS + DELTAY
      DMIN = DMIN + DELTAD
      JFLAG = -1
61    DC 60 N = 1,NB
      IF (DMIN.GT.DLARGE(N)) DLARGE(N) =-1.0
C         IF DLARGE(N) FOR NTH BUOY NEGATIVE, THE
C         POSSIBILITY OF DETECTION NO LONGER EXISTS
      IF (DLARGE(N).LT.0.0) GCTO 60
C
C     FLAG SET TC INDICATE THAT THERE ARE MORE BUOYS TO BE TESTED
C
      JFLAG = 1
      IF (DMIN.LT.DSMALL(N)) GOTO 60
      IF(MON(N).EQ.2)GC TC 77                                             571
      IF(NOW(J,N).EQ.0)GC TO 60                                           572
77    C(N) = SQRT((XB(N) - XS)**2 + (YB(N)-YS)**2)
      IF (NPLIND.EQ.0) GCTO 43
      IF (C(N).GE.ANPTS - 1.C) GOTO 6C
      IC = C(N)
      AIC = IC
      PL(N) = PLC(IC+1) +(PLC(IC+2)-PLC(IC+1))*(C(N)-AIC)
      GCTC 46
43    IF (C(N).GT.1.0) GCTO 44
      PL(N) = 66.C
      GCTC 46
44    PL(N) = 17.0*ALOG1C(C(N))+66.0
C
C     COMPUTE SIGNAL EXCESS FOR EACH BUOY
46    SE(1,N) = FMF(N) - PL(N)
      DC 120 NX = 1,NFM
      SE(NX,N) = SE(1,N)+FMD(NX)
C
C     INCREASE FCM BY 2 DB FOR ALERTED OPERATOR
C
      IF (INCALT(NX).GE.J) SE(NX,N) =SE(NX,N)+2.C
120   IF(SE(NX,N).GT.0.0) II(NX) = II(NX)+1
6C    CCNTINUE
```

```
C
121   DO 134 IXY=1,NFM
      IA = II(IXY) + 1
      IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 240
      GO TO (128, 124, 122), IA
C        II .GE. 3
240   HOLD3(IXY)=.TRUE.
      HTM3(IXY) = HTM3(IXY) + STEP
      JJ3(IXY) = 1
C        II .GE. 2
122   HOLD2(IXY) = .TRUE.
      HTM2(IXY) = HTM2(IXY) + STEP
      JJ2(IXY) = 1
C        II .GE. 1
124   HOLD1(IXY) = .TRUE.
      HTM1(IXY) = HTM1(IXY) + STEP
      INCALT(IXY) = J + IFIX(20./STEP)
      IF (JJ1(IXY) .EQ. 1) GO TO 126
      TDM(IXY) = TDM(IXY) + TREAL
      NFD(IXY) = NFD(IXY) + 1
      JJ1(IXY) = 1
126   IF ((IA.LE.0).OR.(IA.GE.4)) GO TO 134
      GO TO (130,130,132), IA
128   IF (.NOT. HOLD1(IXY)) GO TO 130
      NHT1(IXY) = NHT1(IXY) + 1
      HOLD1(IXY) = .FALSE.
130   IF (.NOT. HOLD2(IXY)) GO TO 132
      NHT2(IXY) = NHT2(IXY) + 1
      HOLD2(IXY) = .FALSE.
 132  IF(.NOT. HOLD3(IXY)) GOTO 133
      NHT3(IXY) = NHT3(IXY) + 1
      HOLD3(IXY) = .FALSE.
 133  IF(JFLAG.LT.0) GOTO 69
134   CONTINUE
C
   50 CONTINUE
69    DO 137 JXY = 1,NFM
      IF (.NOT. HOLD1(JXY)) GO TO 51
      NHT1(JXY) = NHT1(JXY) + 1
      IF (.NOT. HOLD2(JXY)) GO TO 51
      NHT2(JXY) = NHT2(JXY) + 1
      IF (.NOT. HOLD3(JXY)) GO TO 51
      NHT3(JXY) = NHT3(JXY) + 1
51    CONTINUE
      IF(JJ3(JXY).EQ.1)GO TO 55
      IF(JJ2(JXY).EQ.1)GO TO 65
      IF(JJ1(JXY).EQ.1)GO TO 75
      GO TO 137
```

```
 55  AK3(JXY)=AK3(JXY)+1.0/ANRUNS
 65  AK2(JXY)=AK2(JXY)+1.0/ANRUNS
 75  AK1(JXY)=AK1(JXY)+1.0/ANRUNS
137  CONTINUE
140  CONTINUE
     WRITE(6,666)HTM1(1),HTM2(1),HTM3(1)
666  FORMAT(' HTM1 = ',F20.8,'HTM2 = ',F20.8,'HTM3 = ',F20.8)
     DO 155 IXY=1,NFM
     IF (NFD(IXY) .EQ. 0) GO TO 145
     TDM(IXY) = TDM(IXY)/FLOAT(NFD(IXY))
145  IF (NHT1(IXY) .EQ. 0) GO TO 155
     HTM1(IXY) = HTM1(IXY)/FLOAT(NHT1(IXY))
     IF (NHT2(IXY) .EQ. 0) GO TO 155
     HTM2(IXY) = HTM2(IXY)/FLOAT(NHT2(IXY))
     IF (NHT3(IXY) .EQ. 0) GO TO 155
     HTM3(IXY) = HTM3(IXY)/FLOAT(NHT3(IXY))
155  CONTINUE
     RETURN
     END
```

BIBLIOGRAPHY

1. Brownless, K.A., Statistical Theory and Methodology in Science and Engineering, John Wiley and Sons, Inc., 1960.

2. Hoel, P.G., Port, S.C., Stone, C.J., Introduction to Probability Theory.

3. Naval Air Development Center, Functional Specification, TASDA III, Program for Optimal Geometric Selection, Cole, S.N., International Business Machine Co. Inc., contract N62269-72-C-0402, 8 June 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor D.E. Harrison, Jr., Code 61Hx Department of Physics and Chemistry Naval Postgraduate School Monterey, California 93940	3
4.	Assoc. Prof. R.N. Forrest, Code 55Fo Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Chief of Naval Personnel Pers 11b Department of the Navy Washington, D. C. 20370	1
6.	Department of Operations Research and Administrative Sciences, Code 55 Naval Postgraduate School Monterey, California 93940	1
7.	Mr. M. L. Metersky Naval Air Development Center Warminster, Pennsylvania 18974	1
8.	Central Files Naval Air Development Center Warminster, Pennsylvania 18974	1
9.	Cdr. John R. Bliss, USN (student) Commander Third Fleet % F. P. O. San Francisco 96610	1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Modification of the TASDA Computer Program for Inflight Use		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1973
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Robert Bliss		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1973
		13. NUMBER OF PAGES 195
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

TASDA
Sonobuoy Patterns
Simulation
Computer Simulation
A.S.W.
Airborne A.S.W.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

TASDA, acronym for Tactical Airborne Sonar Decision Aid is a computer simulation designed to select optimum sonobuoy pattern spacings given environmental parameters and submarine mode of operation. The program was designed to operate in a Tactical Support Center for briefing of flight crew personnel. Analytical methods and statistical models are used to

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20)

investigate the TASDA program with a view towards modifying it for future aircraft inflight utilization. Some improvements are made to the TASDA model which reduce program run time and core storage requirements. A modified version of the TASDA program is developed as an initial step toward an inflight model.

Thesis 146875
B564 Bliss
c.1 Modification of the
TASDA computer program
for inflight use.